NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A PROTOTYPE DATABASE MANAGEMENT SYSTEM FOR THE BUDGETING SYSTEM OF THE DEPARTMENT OF DEFENSE AND SECURITY OF THE REPUBLIC OF INDONESIA

by

Mohammad Subekti
and
Widhya Bagya Prawiraatmadja

September 1985

Thesis Advisor: Michael P. Spencer

Approved for public release; distribution is unlimited

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*: A Prototype Database Management System for the Budgeting System of the Department of Defense and Security of The Republic of Indonesia	5. TYPE OF REPORT & PERIOD COVERED: Master's Thesis, September 1985
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Mohammad Subekti, Widhya Bagya Prawiraatmadja	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, CA 93943-5100	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, CA 93943-5100	12. REPORT DATE: September 1985
	13. NUMBER OF PAGES: 127
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

The Prototype DB for Budgeting System

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The management in the Department of Defense and Security of the Republic of Indonesia (DODS) needs relevant, up-to-date information in query type processing to manage the Budgeting System in the DODS. The budget planning cycle and the budget management cycle are presented briefly in order to define the system requirement for the budgeting system in the DODS. Based on that requirement the discussion of the data base management system (Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

ABSTRACT (Continued)

includes the general structure of data, the impact of the data base development to the DODS management, and a cost benefit analysis concept.

Approved for public release; distribution is unlimited.

A Prototype Database Management for the Budgeting System of The Department of Defense and Security of The Republic of Indonesia

by

Mohammad Subekti
Captain (E), Indonesian Navy
B.E., Naval Electronic School, 1972

and

Widhya Bagya Prawiraatmadja
Captain (Inf), Indonesian Army
Armed Forces Military Academy, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

ABSTRACT

The management in the Department of Defense and Security of the Republic of Indonesia (DODS) needs relevant, up-to-date information in query type processing to manage the Budgeting System in the DODS. The budget planning cycle and the budget management cycle are presented briefly in order to define the system requirement for the budgeting system in the DODS. Based on that requirement the discussion of the data base management system includes the general structure of data, the impact of the data base development to the DODS management, and a cost benefit analysis concept.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

Professor Michael Spencer, Professor Richard McGonigal, Professor Daniel Dolk, and Professor Carl Jones have contributed substantially to the completion of this thesis. We would like to express our great appreciation to those gentlemen publicly.

We also would like to thank to Mrs. Nicole Feurer, Mrs. Marjorie Crawford, Lieutenant Commander Billie Crawford, and Lieutenant Diane Gifford for helping us in editing of this thesis; and Colonel Amin Muhadi, Lieutenant Colonel Yahya Arifin, and Major M. Rosid for their contribution in providing the reference books about Indonesian Government and DODS Budgeting System.

Most importantly, we thank our wives, Mrs. Nani Subekti and Mrs. Enny Prawiraatmadja S.H., without whose constant support this thesis may never be have been completed.

I. INTRODUCTION

The Budget Planning Cycle and the Budget Management Cycle play an important role in the management system of the Department of Defense and Security of the Republic of Indonesia (DODS). The Budget Planning Cycle controls the process of allocation in the budget for funding the DODS activities which have to be done in the forthcoming fiscal year. On the other hand the Budget Management Cycle controls the implementation of the budget to ensure the budget will be spent effectively, efficiently and legally. These two cycles are currently processed in the computer using the file processing mode. According to David Kroenke [Ref. 1], the file processing mode has many disadvantages compared to the database processing mode. It takes a lot of time, has a lot of uncontrollable data redundancy which may lead to a leak in data integrity, and has slow response to new and unpredictable requirements. Any new requirement may require program modification or new program development and new data structure, which afterwards may require modification to all of the developed programs. Chapter II discusses and defines the current management problems in the Budget Planning Cycle and the Budget Management Cycle.

Chapter III will discuss the general concept of the database management and the database structure to be used in the Budget Planning Cycle and the Budget Management Cycle. This concept is independent of particular hardware and software to be used.

Any system development will always require some modifications such as hardware devices, quality of personnel etc. These requirements are discussed briefly in Chapter

IV which describes the management impact of the Data Base Management System (DBMS) concept set forth in Chapter III and how it will be implemented.

Analysis of the cost and benefit to the DBMS concept described in Chapter III is discussed briefly in Chapter V. This chapter discusses the advantages and disadvantages of the concept compared to the current system.

The final chapter comprises the content of all previous discussions and describes recommendations to the DODS management, which the authors feel are essential to ensure implementation of the database management and the database structure concept to comply with the requirement of the system.

II. REVIEW OF REQUIREMENT ANALYSIS

A. BUDGET PLANNING CYCLE

The Budget Planning Cycle is a set of processes which control the activities in DODS to determine which activities and projects have to be done and how much money must be allocated to those activities and projects for the forthcoming fiscal year [Ref. 4]. These processes include priority setting, determination of which units are to be responsible to carry out the activities or implement the projects, and the policy of how the budget will be expended.

The flow of the Budget Planning Cycle in the Indonesian government is shown in Figure 2.1.

Every 30th of September, DODS must submit the List of Proposal Activities (LPA) and List of Proposal Projects (LPP) to the Department of Finance. The Department of Finance, as coordinator of the budget preparation will then collect and integrate all the LPAs and LPPs from every department and other non departmental government institutions. The integrated LPAs and LPPs will then be submitted to the House of Representatives after its has been signed by the President and becomes the Government Budget Proposal. The submission under normal circumstances will be made at the end of December. From those LPAs and LPPs then the House of Representatives together with the government will evaluate the Government Budget Proposal. After the government and the House of Representatives come to an agreement, the government will announce it as the Government Budget for the forthcoming fiscal year. The announcement is scheduled for the first of April. If the House of Representatives and the government cannot reach an agreement

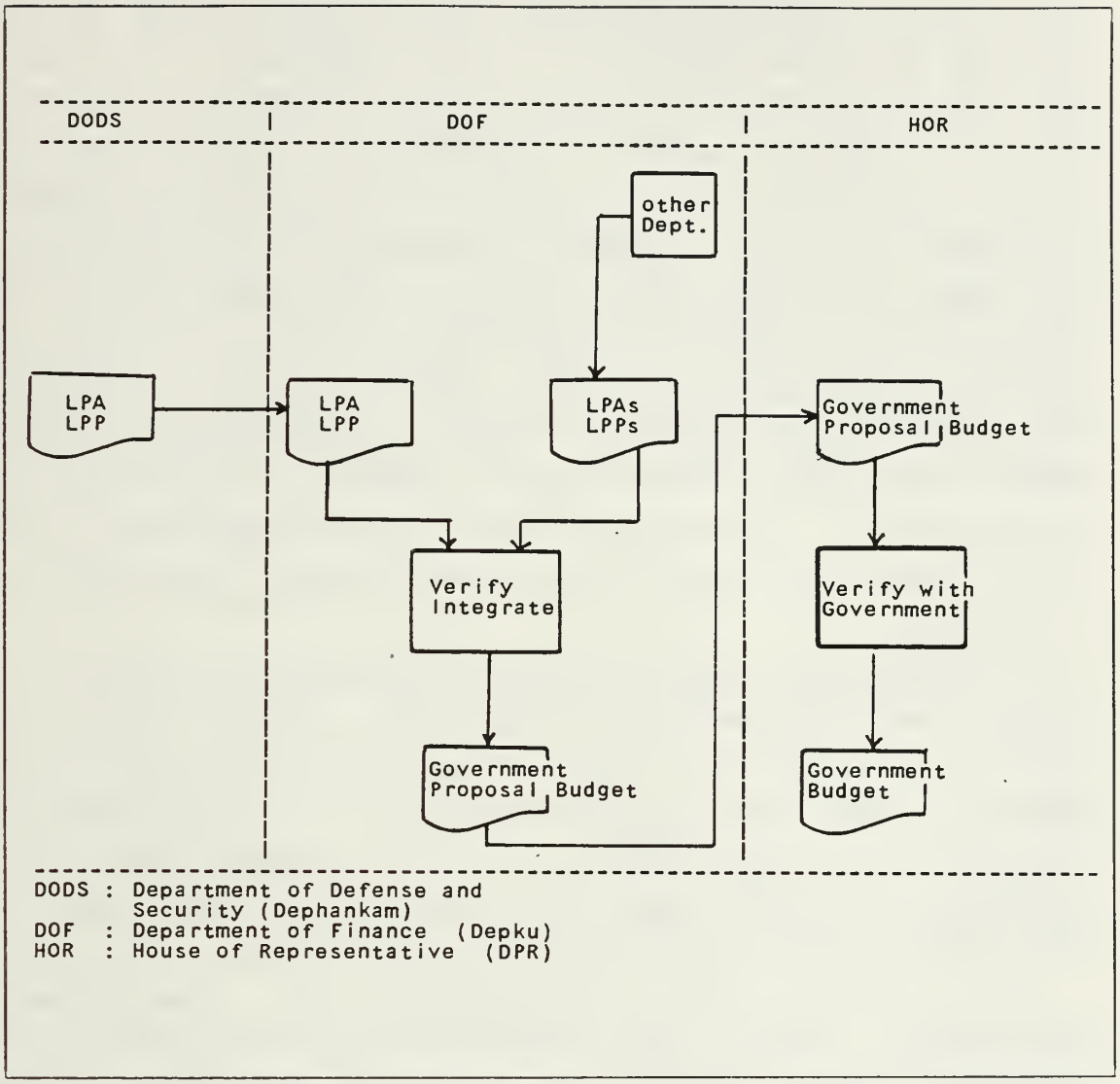

Figure 2.1 The Budget Planning Cycle.

by the first of April then the government must use the last budget plan.

DODS must start the budget preparation activity at least 2 months in advance to enable submitting the LPA and LPP to the Department of Finance by the 30th of September. An advance preparation is necessary because the process of collecting input data and the computer processing usually takes at least two months.

The main problem for the Budget Planning Cycle in the DODS is the negotiation process which occurs after the LPA and LPP have been submitted to the Department of Finance. When the Department of Finance receives the LPA and LPP from DODS, it usually does not directly agree with the amount of budget proposed by the DODS. Most of the time the Department of Finance will ask the DODS to modify and lower the budget according to their prediction of how much budget the government can allocate to DODS for the coming fiscal year and the prediction of how much budget will be excepted by the House of Representative. This activity has to be done before the LPAs and LPPs are submitted to the House of Representative. For there the negotiation session will start on the first of October and must be completed before the end of December.

There are two types of negotiations which always occur during the negotiation process. The first one is the external negotiation. This negotiation occurs between the managers of the DODS and the managers from the Department of Finance. The main purpose of this negotiation is to determine how much money can be allocated to the DODS and the considerations of that allocated budget.

The second type of negotiation is the internal negotiation. This type of negotiation occurs between the budgeting manager and the other functional manager in the DODS. After the allocation of the proposed budget has been discussed in the external negotiation, then the result must be discussed internally between the functional managers in DODS to determine which activities have to be cancelled or which projects have to be postponed.

Those two types of negotiation may occur many times. Every negotiation may come up with a different amount of budget allocation. To meet the amount of proposal budget which has been determined during the external negotiation,

some of the activities must be reanalyzed and readjusted and the related projects as well. To understand how the proposed budget will be modified we have to know how the budget is organized. This is described in the last section of this chapter.

The database management system can be used to expedite the modification process in the Budget Planning Cycle and not only this, but the database management system can also be used to help the managers during the negotiation process, because the database management system is ideal for the decision support systems. How this can be done is discussed in the next chapter.

B. BUDGET MANAGEMENT CYCLE

The Budget Management Cycle is a set of processes which control the implementation of the budget during the fiscal year. These processes include the control of the authorization flow, the funding flow and the finance report required by law [Ref. 4].

There are three types of data flows in the Budget Management Cycle : Authorization Flow, Funding Flow and Money Transfer (see Figure 2.2.).

To be able to use the budget each authorized government official must have an authorization letter from his or her superior. The Minister of DODS receives the Authorization Letter from the President through the Minister of Finance, usually every three months. The authorization letter contains the main programs that must be accomplished, the organization (in this case DODS), the amount of budget detailed in each major type of expense and some other necessary descriptions, such as changes made to the budget. The Minister of DODS then distributes the authorization to the Chief of Staff of each branch. The content of the

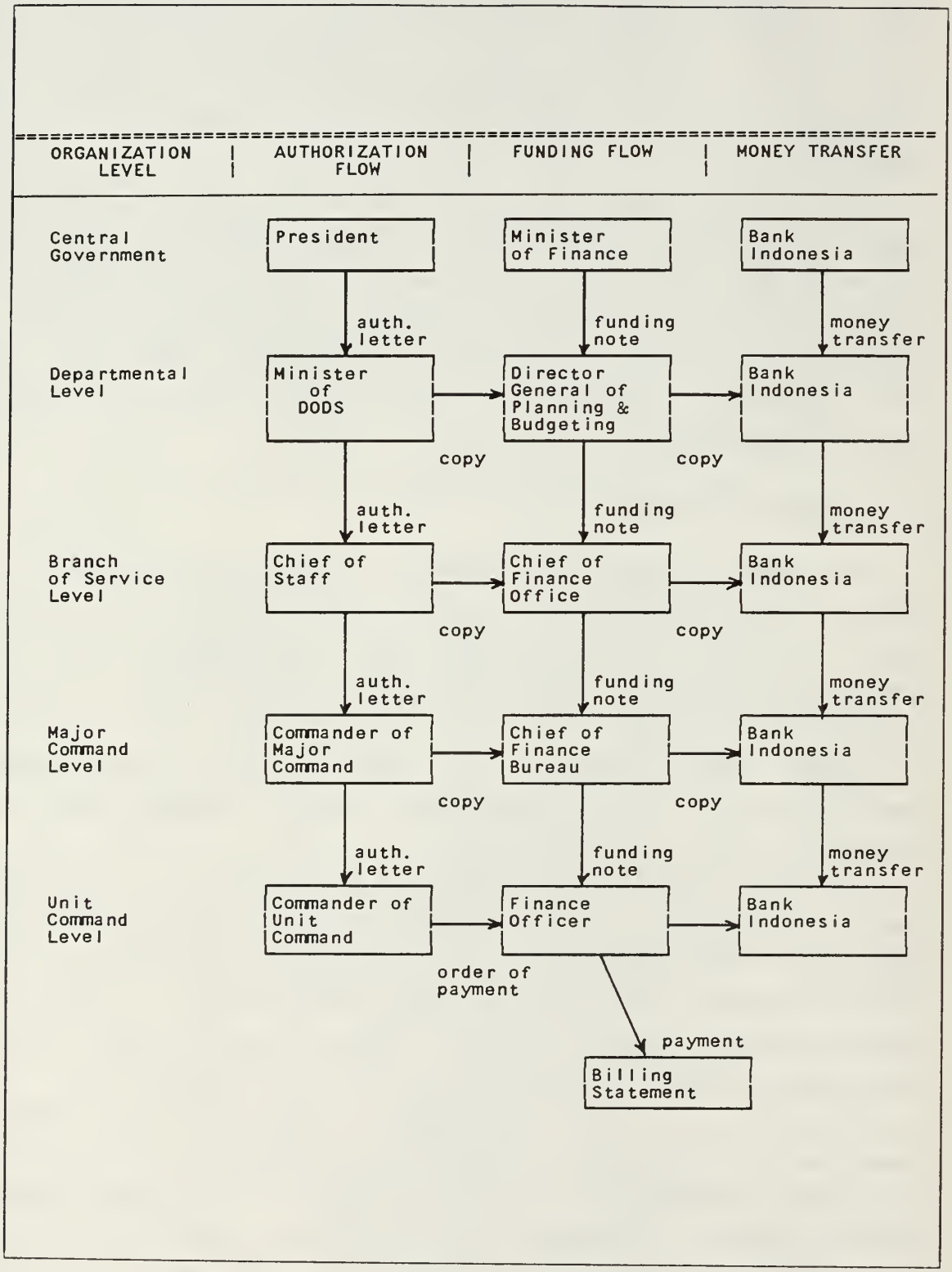

Figure 2.2 The Authorization Flow, Funding Flow, and Money Transfer.

Authorization Letter from the Minister of DODS to the Chief of Staff of each branch is more detailed than the authorization letter from the President to the Minister of DODS. It describes which programs must be accomplished by the branch, the organization (in this case the branch), the amount of budget detailed in each submajor type of expense and other management guidance. The next level of official government in DODS who receives the Authorization Letter is the Commander of the Major Command. Each Commander of the Major Command receives their budget authorization from their Chief of Staff in their branch. The contents of the Authorization Letter at this level are the activities which must be done by the Major Command, the unit expense of the budget, and the amount of the budget. The last level of the official government in DODS who receives the authorization letters is the Commander of the Unit Command. They receive their Authorization Letter from their Commander of the Major Command.

The funding flow is used by the financial managers to inform them that the budget has been transfered into their account. The Director General of Planning and Budgeting of DODS receives the Funding Note from the Minister of Finance usually every three months. The Funding Note contains the authorization number, the amount of budget, and the bank account number of the sender and the receiver. The Director General of Planning and Budgeting then distributes the fund to every Chief of Finance Office of each branch. The Chief Finance Office then distributes to every Chief of Finance Bureau in every Major Command. The last official government level of financial managers who receive the Funding Note are the Financial Officers in every Unit Command.

The bank transfers the money from one account number to another account number through the Bank Transfer Note. All the government organizations must have a bank account in the

government bank (Bank Indonesia), so there are no transfer charges for transfers, but the money also receives no interest.

The Commander of the Unit Command, every time he or she wants to use the budget to pay a bill must issue an Order of Payment. The finance officer will pay the money after the Order of Payment Letter has been checked against the Letter of Authorization received by the Commander of Unit of Command to insure that the payment is legal. The Order of Payment letter contains the Authorization Letter number, the company or person who is entitled to receive the money, the budget codes and the amount of money to be paid.

The managerial problems of the budget management cycle is the monitoring of these three flows: the authorization flow, the funding flow and the bank transfer. Every manager at every level must keep track of : which Authorization Letter has not been funded, how much money has been authorized, for what purpose, how much has been transfered and to which bank account, how much has been spent and how much is left, and which activity has to be funded first and when. Those are the daily problems of each financial manager at every level.

The database management system is ideal for meeting those requirements. Using the database management system, the user can easily refer to any data in the database according to any order. The database can also be organized in order to protect a certain group of data or data structure to ensure that only an authorized person or group can access the data. This feature is ideal for the superior who wishes to protect certaininformation which he does not want to be seen by his subordinate or for updating of data which can only be done by an authorized person.

C. BUDGET STRUCTURE

The budgeting system used by the DODS must follow the government budgeting system [Ref. 4], so the budget structure of the DODS must be aggregated according to the budget structure of the government budget.

1. Structure of the Government Budget

The budget structure of the government budget is divided into two types of budgets. The first is the Maintenance Budget ('Anggaran Rutin') and the second is the Development Budget ('Anggaran Pembangunan'). The Maintenance Budget is used to support the substantial routine activities; and the Development Budget is used to support the government development programs.

These two types of budget then must be able to be analyzed according to the expense budget. There are four types of major expenses. The first is the Personnel Expense Budget which is used to aggregate all the personnel expenses such as payroll, position allowance, housing allowance etc. The second is the Maintenance Expense Budget which is used to aggregate all the maintenance expenses such as vehicular maintenance, weapon maintenance, office maintenance, government housing maintenance etc. The third is the Procurement Expense Budget which is used to aggregate all the procurement expenses such as gasoline, electricity, water, gas, ammunition etc. The last is the Transportation Expense Budget which is used to aggregate all the transportation expenses such as moving expenses, TDY etc [Ref. 5].

Each major expense then can be detailed into submajor expense, and each submajor expense then can be detailed into unit expense. The Expense Budget Structure then can be visualized in Figure 2.3. The first level of the

expense structure is the Major Expense. The second is the Submajor Expense, and the last is the Unit Expense.

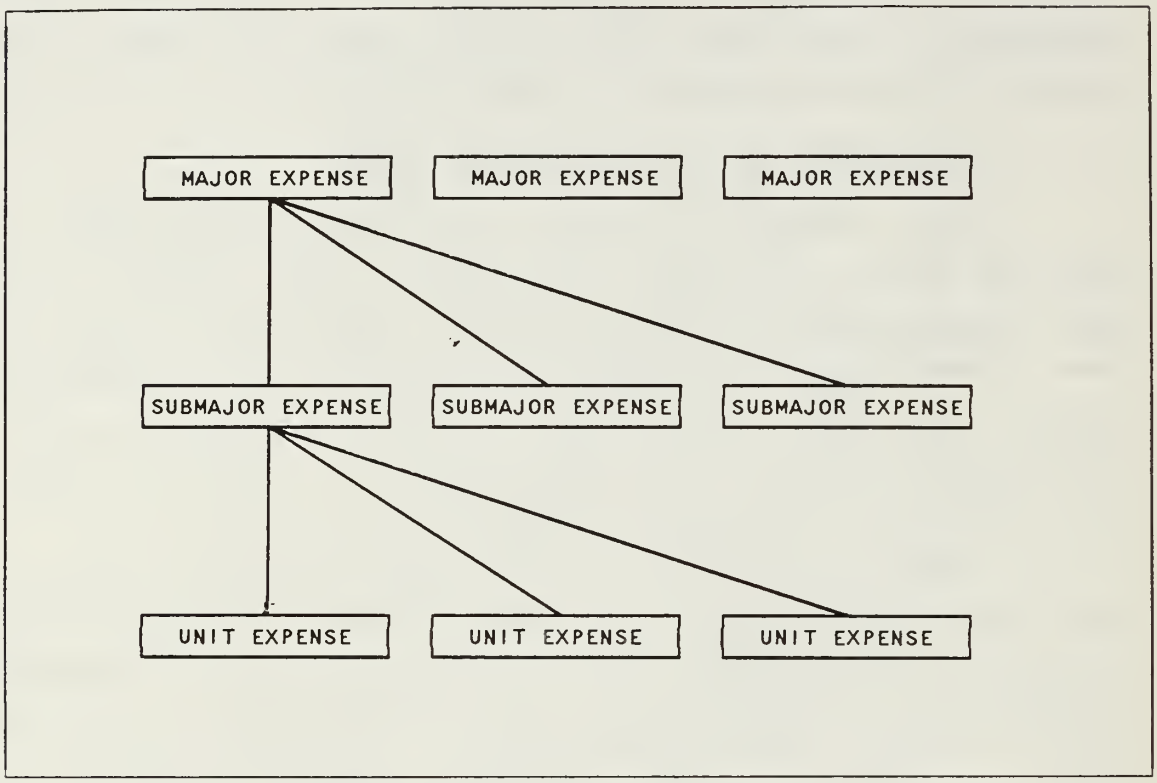

Figure 2.3 The Expense Budget Structure.

Those two types of budget (maintenance budget and development budget) must also be recognized according to the government missions. The first aggregation mission is the Sector which describes the area of the government mission such as the agriculture sector, manpower sector, defense and security sector etc. The Sector is then divided into Main Programs. There are four Main Programs for the Defense and Security Sector: the Defense Forces Main Program, the Security Forces Main Program, the Administration and Management Main Program, and the Bhakti ABRI Main Program. Each Main Program consists of many Programs. An example of

the Defense Forces Main program would be Territorial Defense Forces Program, Strategic Defense Forces Program etc. Each Program contains a certain number of activities. The Activity is the building block of the program in which the program is actually measured and accounted. Figure 2.4 describes the Program Structure of the government budget. The first level is the Sector, the second is the Main Program, the third is the Program and the last level is the Activity.

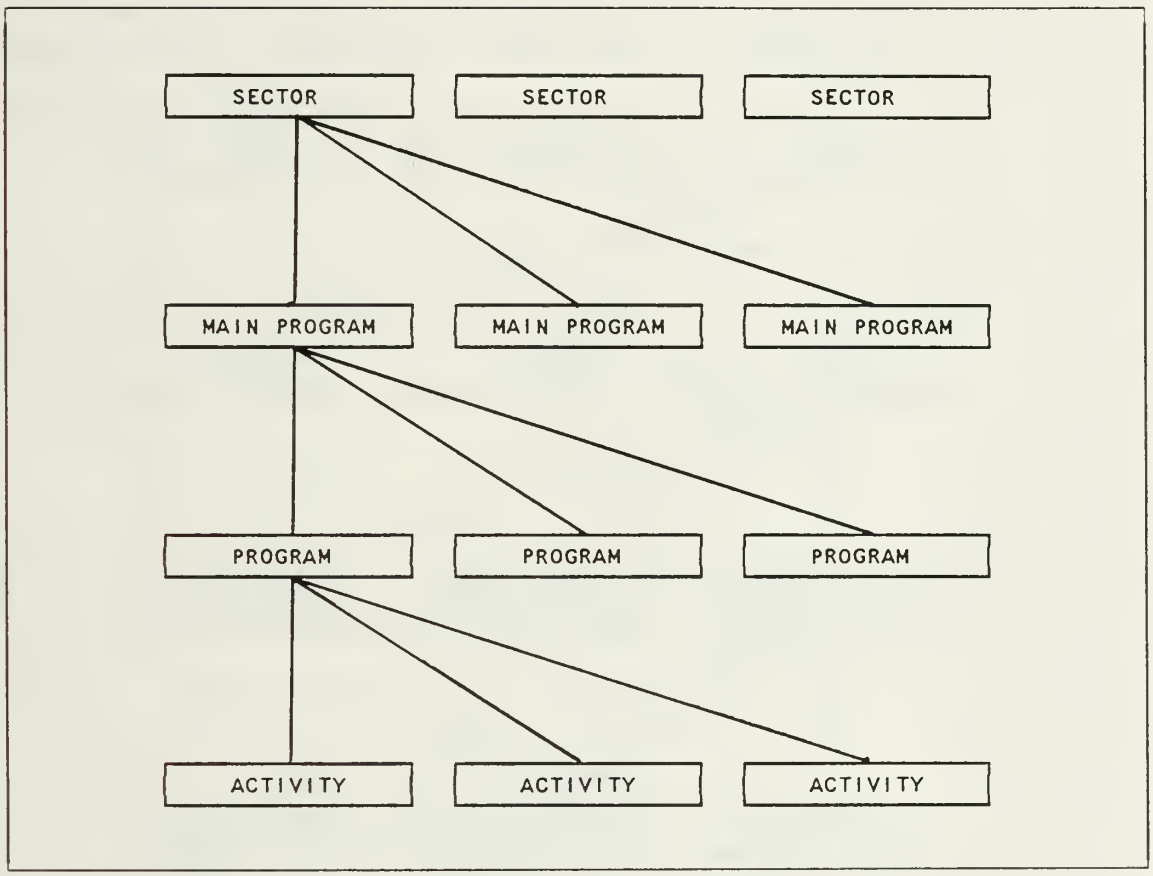

Figure 2.4 The Program Structure.

The budget is allocated according to the recognition of the organization which preseats its budget. The highest

organization recognized in the government budget is the Department which is lead by a Minister. The lowest organization recognized by the government budget is third level under each department. Every department has a different name for each level according to its specialty. In DODS the first level down after the department is the Branch of Services, the second level is the Major Command and the last is the Unit Command. Figure 2.5 describes the structure of the organization in the DODS for budgeting system purposes.

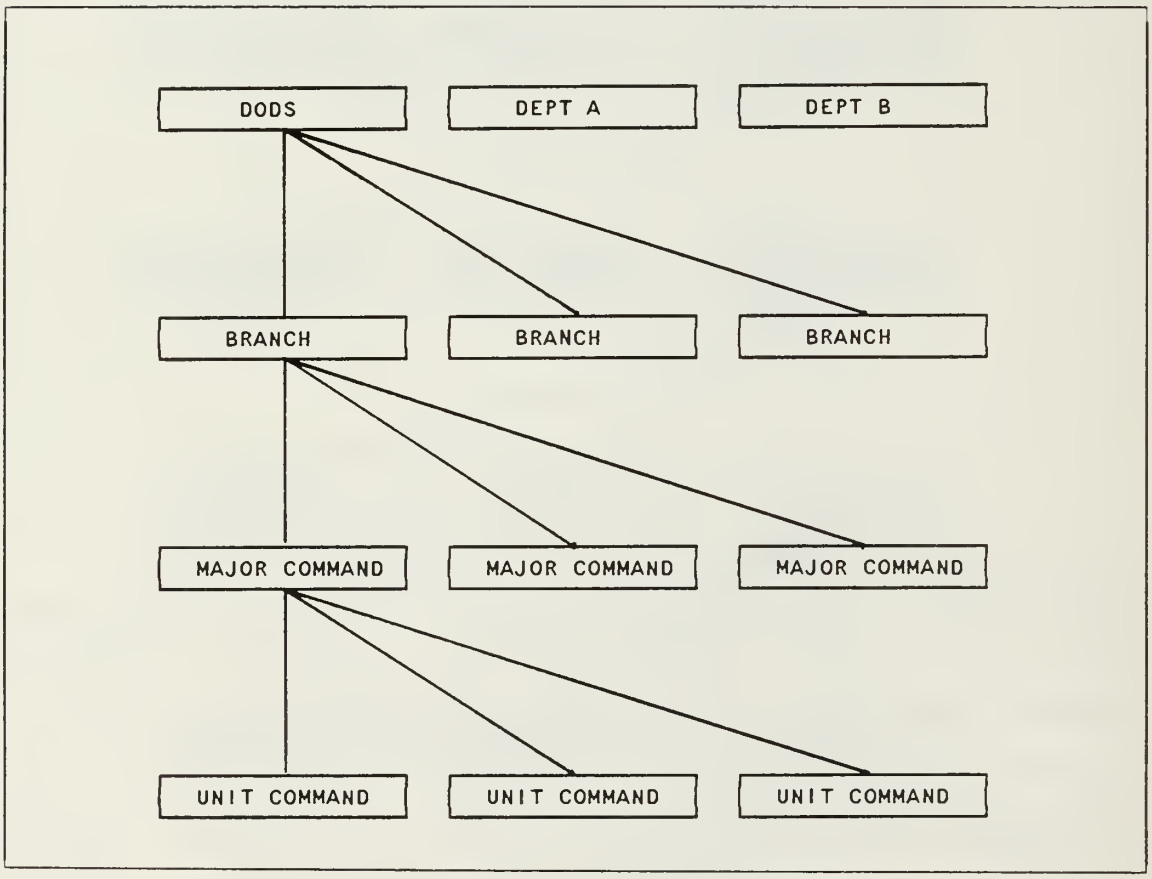

Figure 2.5 The Organization Structure.

The government budget also recognizes a centralized fund budget for certain budget entries such as the budget for the gasoline, electricity, telephone bill, rice, etc. This budget will still allocate to the appropriate organization, but the fund will be kept in the Department of Finance and the organizations will receive the material or goods or service instead of the money itself.

So the whole picture of the Government Budget Structure can be visualized as is seen in Figure 2.6. The budget must be able to be recognized according to the Type of Budget, the Expense Structure, the Program Structure, the Organization Structure and the Centralized or Non Centralized fund [Ref. 5].

2. Structure of the DODS Budget

The structure of the DODS budget is similar to the structure of the government budget. The main difference is only in the scope. The DODS budget is a part of the government budget, so the scope is smaller. But because it is smaller, it is more detailed than the government budget.

In addition to the government budget, in DODS the budget must also be able to be recognized according to the budget fiscal year. There are three different types of budget according to the fiscal year; the current fiscal year budget, the remainder from the previous fiscal year budget, and the addition for the current fiscal year budget. The remainder from the previous budget consists of money which has not yet been used because of certain situations occurring such as the budget for a project which has to be postponed because of a catastrophic situation. The additional budget for the current fiscal budget is the budget given by the government in a special situation such as for an emergency situation which needs more budget support and the current budget is not enough to handle it.

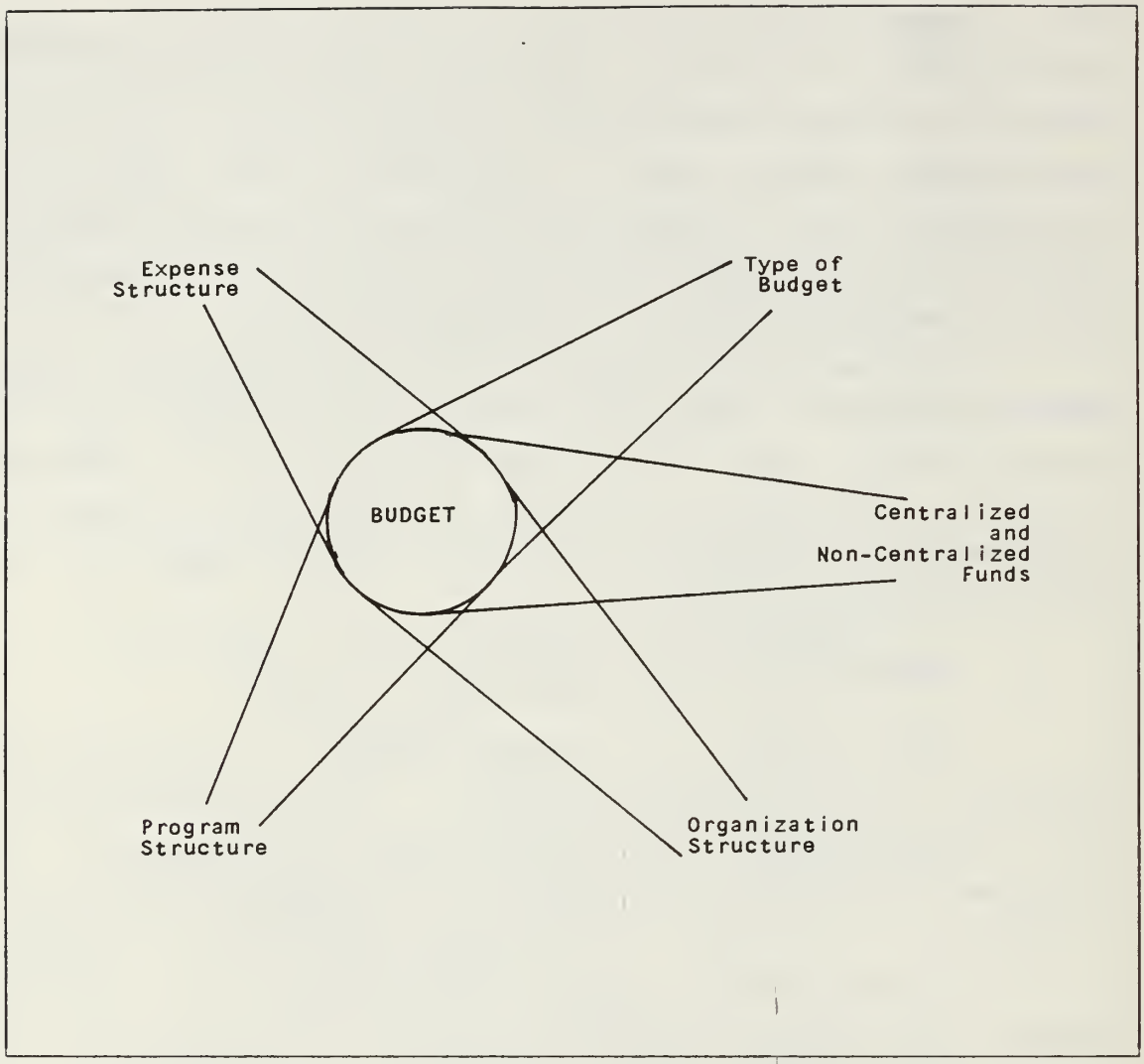

Figure 2.6 The Government Budget Structure.

The DODS budget structure also recognizes the centralized fund for a certain budget entity. The centralized fund budget is also still to be allocated to each branch but the fund will be kept in the DODS. The DODS office will take care of the payment and the branches will receive the goods. Usually the budget measured in foreign currency such as for the weapon procurement from a foreign country will be centralized in the DODS office.

Another additional structure for the budget is the control program and the supervision program. Every program should have a control program and a supervision program. The supervision program is used to insure that each activity in the program moves toward the same goal and that the movement of the activities in a program are synchronized with each other so there is no fraction or overlap between activities in a program. The control program is used to ensure the synchronization of program movement. Every program should move in a synchronized fashion toward the defense and security goals.

The overall picture of the DODS Budget Structure then can be visualized in Figure 2.7. The budget has to be recognizable by the type of budget, the program structure, the organization structure, the expense structure, the control program and supervision program, the centralized fund in the Department of Finance and the centralized fund in DODS, and the division according to the fiscal year [Ref. 5].

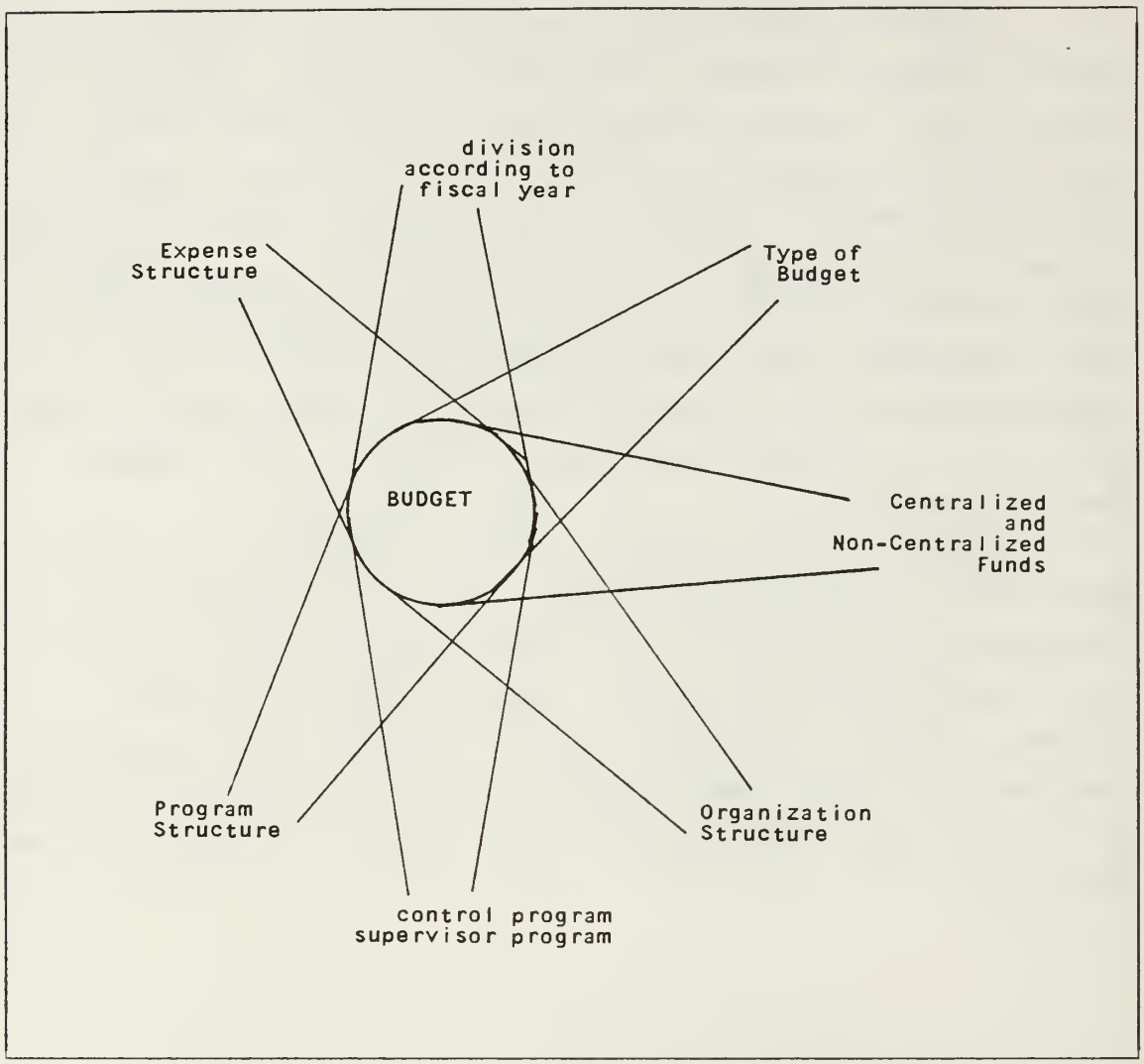

Figure 2.7 The DODS Budget Structure.

III. DATABASE MANAGEMENT TO SUPPORT DODS BUDGETING SYSTEM

A. INTRODUCTION

This chapter will show how the Database Management System (DBMS) can be used to provide a better solution for managerial needs in the DODS Budgeting System. In this case the authors propose a prototype of the Database Budgeting System for two reasons. First, due to the distance between authors as developer and the actual users of the DODS Budgeting System, it is difficult to establish an effective communication. Therefore, some methodology such as interview, observation, and brainstorming which involves the users has not been conducted. However, since one of the authors was actively involved in the developing and maintaining of the present system, we can assume that the authors can understand the actual requirement, the unsolved problems in the present system, and the future needs of the users and the system.

Secondly, prototyping is widely used in many cases of software development and successfully implemented by many organizations. It has been proven [Ref. 3] to be the most cost effective in developing computer application software where the degree of computer knowledge of the user is very low. It allows the software to be reviewed by users, designers, and managers and then after several iterations a running version of the system can be obtained.

This chapter will discuss how to model the DODS Budgeting System which involves the environment and data processing in order to define what kind of data will be stored in the database. In principle, we would not be able to store all fields in the database system because the

system will suffer in the performance and maintenance effort. Therefore, in the process of developing the prototype database for DODS Budgeting system, the authors will be concerned with aggregating data as much as possible, combining data into higher levels and generalizing data, ignoring the differences between data as much as possible. The aggregation and generalization of data is done without losing any information that might be needed by the users.

After all possible data to be kept in database can be identified, the next step is to group such data fields into logical database records with regard to the possibilities of modification anomalies, data duplication, and data redundancy problems. This is a very critical step, because any mistake can result in performance penalties and data integrity problems. In the design of the logical database record, one must also consider the maximizing of processing efficiency with regard to user requirements in terms of the query analysis, output format, and retrieval update.

Finally, the logical design is followed by designing the record relationship to support user access to the database system, processing efficiency, and technological availability. The design must consider the relationship between records which could be one-to-one, one-to-many, or many-to-many.

B. DATA IDENTIFICATION

In order to develop a Database for the DODS Budgeting System, the first step is to identify data to be stored in the database. This is the first step of the Logical Database Design.

There are two considerations involved in the logical design including the Budget Planning Cycle and the Financial Management Cycle. The Budget Planning Cycle focuses on the

process of the Budget Allocation to support the DODS and Armed Forces activities for the following fiscal year. On the other hand, the Financial Management Cycle focuses on the implementation and transaction reporting for the allocated budget. In the current system, these two cycles are separated from each other and are handled by different file processing systems. Therefore, very limited information can be obtained from the system, it is difficult to be analyzed, and does not support the management in the decision making process.

For that reason, the authors propose the new budgeting system be an integrated one. We will use the context diagram, as in Figure 3.1, to show the relationship between components and to identify the document flow in the system.

1. Data Dictionary

The initial input to the current Budgeting System is data taken from Personnel and Material computerized application systems. Various formats of the file are reformatted and restructured into a single format, producing a file called 'FORCE_STRENGTH'. These files are collected from many levels of the organization, depending on what organization the particular application is processed in. The file may be taken from the DODS Computing Center, the Branch of Service Computing Center, or the Major Command level. As other inputs, for the non-computerized data, the Unit Command sends the transaction document called P1 through P13 forms, which can be seen in the [Ref. 6]. These documents consist of FORCE_STRENGTH data and LUMPSUM data. The document refers to the POLICIES and NORM_INDEX given by the DODS Directorate General of Budgeting and General Planning.

The combination of FORCE_STRENGTH data, LUMPSUM data, POLICIES, and NORM_INDEX produce an initial Budget

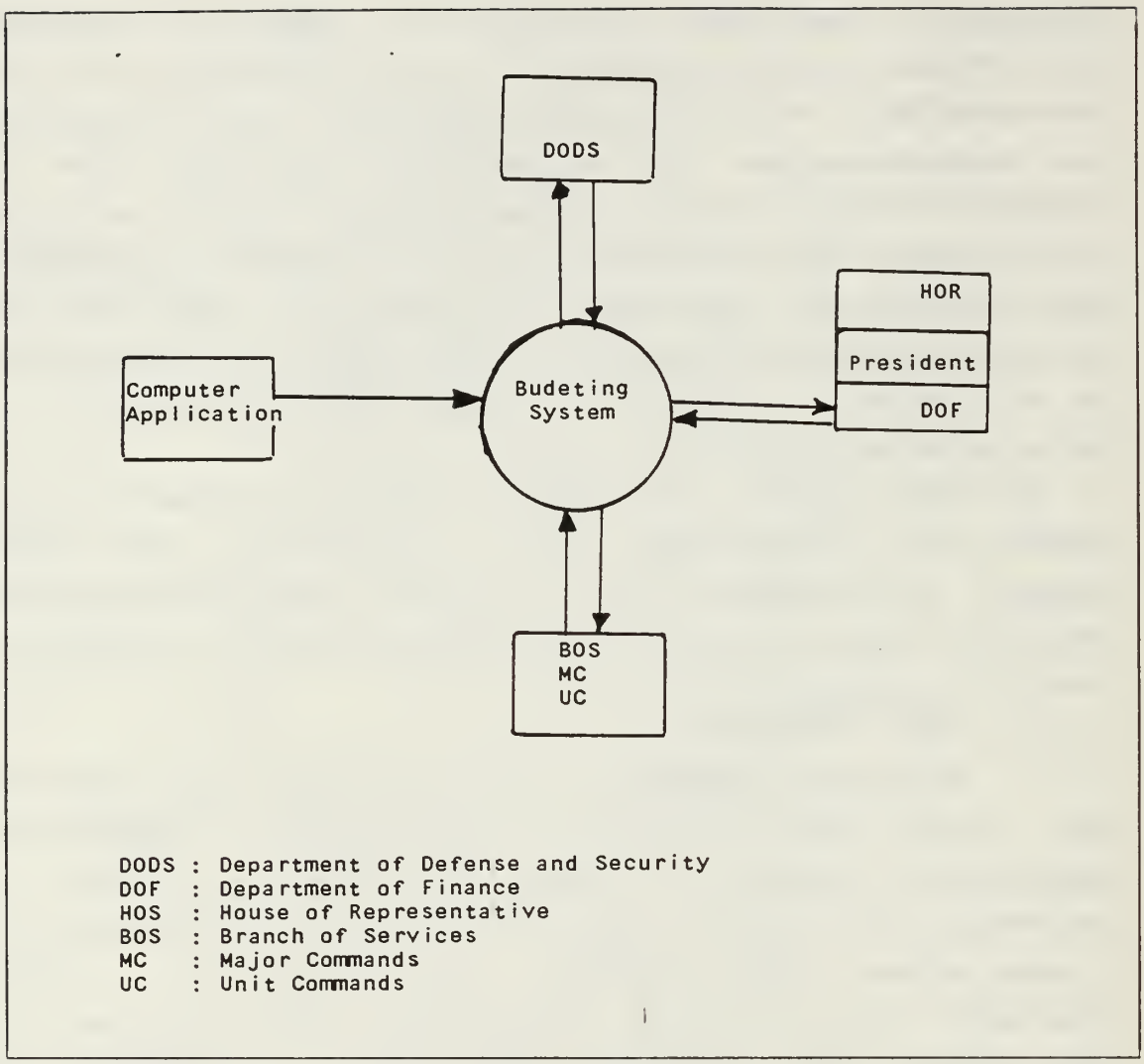

Figure 3.1 Context Diagram of DODS Budgeting System.

file that will be used as input for the negotiation process. The FORCE_STRENGTH, LUMPSUM, and NORM_INDEX can be changed during the negotiation processes. Output Reports produced from that file will go back and forth through the organization until the final agreement can be obtained.

The Proposed Budget will become a legal document containing the summary and detailed report of the Budget items. This document will be distributed to the Department

of Finance, the Department of Defense and Security and the overall lower levels of the organization in the form of a Budget Allocation. The Budget Allocation is used as a base document for every level of the DODS Organization to establish their activities. It cannot be changed, except in special cases that need to be approved by the House of Representatives and the President.

In the Budget Implementation phase, the Financial Management Cycle, every quarter the Department of Finance will distribute the three types of documents named the Authorization Flows, the Funding Flow, and the Money Transfer to each department. These three documents, will be further spread at each level of the organization until they reach the appropriate level. The lowest organization level in the DODS is Unit Command. Based on these documents, the Unit Command establishes payment for any bill that satisfies the criteria for acceptance by the Budget Allocation. Every transaction must be recorded and reported to the upper level of the organization. And finally, it will be processed by the Data Processing Center and become a part of the Financial Responsibilities Documents.

As final output, when every organization agrees with the content of the Budget, it becomes a Budget Proposal that will be sent to the House of Representatives through the Department of Finance. It will be signed by the President prior to its submission to the House of Representatives.

In the requirement analysis stage, it is generally difficult to tell exactly which data is really needed in the database [Ref. 3]. Referring to the Budget Structure mentioned in Chapter II, we will be able to identify some data fields that belong in the system. We will collect these data fields into an initial data dictionary on which each data element will be described as the following:

a. Name

The data field name is the unique name given to the data field. This unique name can be used to trace a field throughout the system. For example: BRANCH_OF_SERVICE

b. Description

The data field description provides a narrative explanation about the meaning of the data field name. For example: BRANCH_OF_SERVICE can be defined as the five branches under the DODS, such as DODS Staff, Army, Navy, Air Force, and Police.

c. Format

The data field format defines whether the content of data field is alphabetical, numeric, or alphanumeric. It also defines the field length. For example, the format of data field BRANCH_OF_SERVICE is alphanumeric with maximum length 30 characters.

d. Coding

If code is used in the data representation in a data field, this coding must be included in the data dictionary. For example, Branch of Service is ARMY, coding is 1202.

e. Addition information

An addition information might be included in the data dictionary such as source of data, where used, storage, and synonym (alias) [Ref. 2]. Figure 3.2 shows the complete example of a data field definition in the data dictionary.

The complete data dictionary identified from data flow analysis of the DODS Budgeting System is listed in Appendix A.

Name	:	BRANCH_OF_SERVICE
Description	:	Define branches under the DODS organization such as DODS Staff, Army, Navy, Air Force, and Police
Format	:	Alphanumeric maximum 30 character length
Coding	:	Code / Name 1201 DODS Staff 1221 ABRI Headquarter 1222 Army 1223 Navy 1224 Air Force 1225 Police
Where used	:	Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.
Storage	:	Budget File
Synonyms	:	Angkatan, Unit Organisasi

Figure 3.2 Example of the DODS data dictionary.

C. LOGICAL DATABASE RECORD

A record is a group of data fields which satisfies the particular criteria. Before a decision is made to assign fields to a record, we would like to try to model the way in which users perceive data. The approach to be used is to analyze the user views on data structure and the queries they might make, the report layouts, and retrieval update of the budget data.

1. Query Type

The query type describes the user view of data. User view can be defined as the smallest set of data elements required to answer a user question which allows the user to make decisions and provides information to the user. Based on the Budget Structure, there are four types of queries

usually asked by the users: single-level single-structure, multi-level single-structure, single-level multi-structure, and multi-level multi-structure.

a. Single-Level Single-Structure Query

This is the simplest user view of Budgeting Data. Following examples of typical questions that indicate this type of query. "How much money is in the total budget for the Army?", "How much money is needed if the personnel strength is increased by 20 %?", etc. This type of query is illustrated in Figure 3.3.

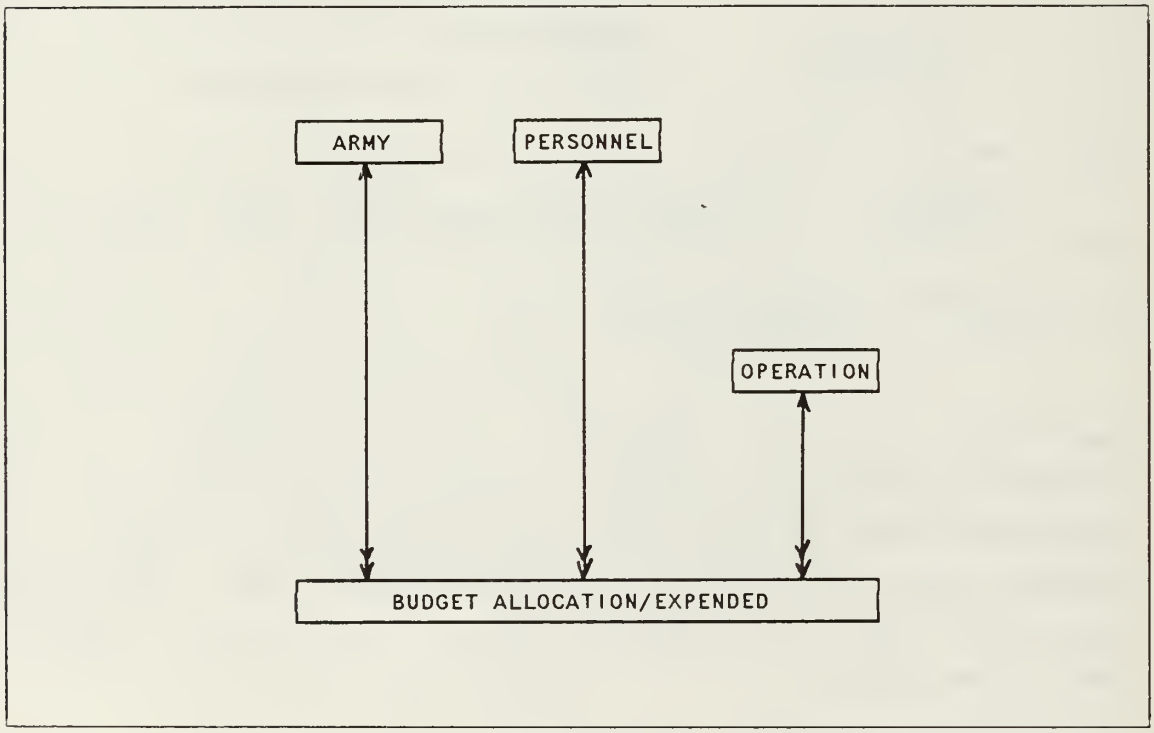

Figure 3.3 Single-Level Single-Structure Query.

b. Multi-Level Single-Structure Query

This query involves accessing data at more than one level within a budget structure. Examples of queries of this type follow: "What is the personnel strength in the Army for Major Command X, Y, and Z?", "What is the total maintenance budget for armed vehicles?", and "What is the total budget expended on the personnel for training?". This query is illustrated in Figure 3.4.

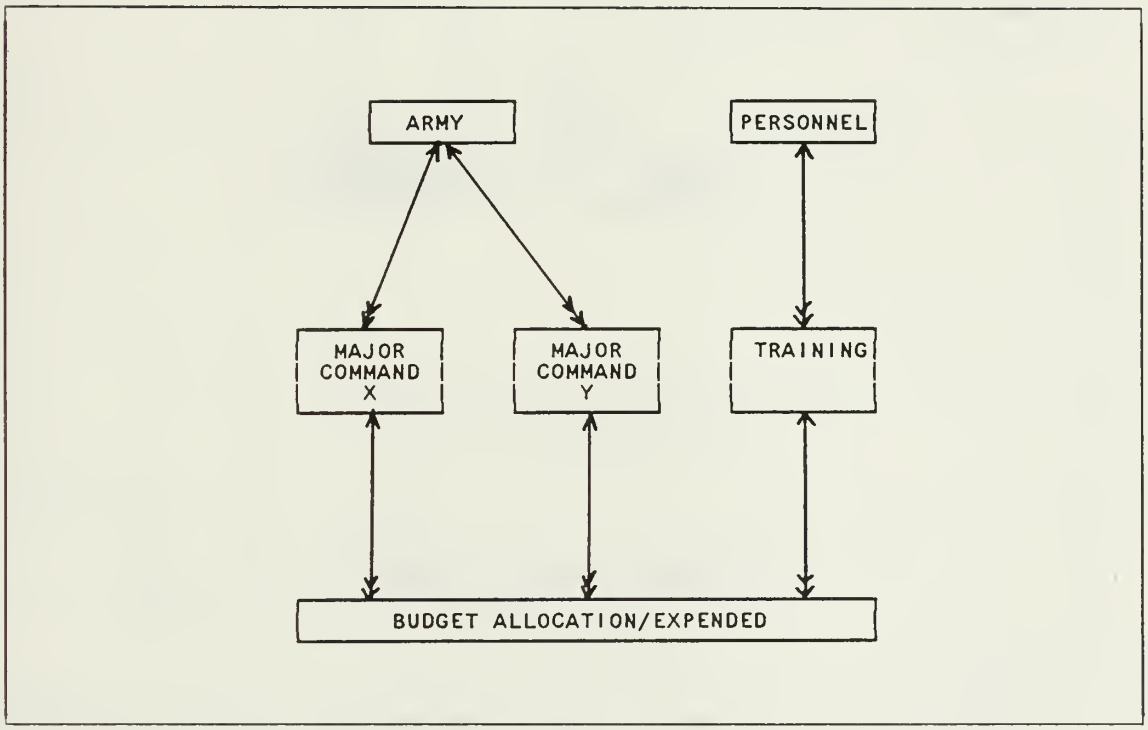

Figure 3.4 Multi-Level Single-Structure Query.

c. Single-Level Multi-Structure Query

This query involves more than one budget structure but only one level of each budget structure. The major difficulties with this kind of query occur because of

the many-to-many relationship between budget structures. In the current system, accessing data like this involves complicated and very tricky programming effort. As an example, for the organization view of budget structure, let's say each Branch of Service has more than one Major Expense, On the other hand, for each Major Expense there is more than one Branch of Service is involved. More complications occur when the lower level of Budget Structure is involved. A typical question in this type of query, "What is the total budget in the Army for Personnel Expenses?". See Figure 3.5.

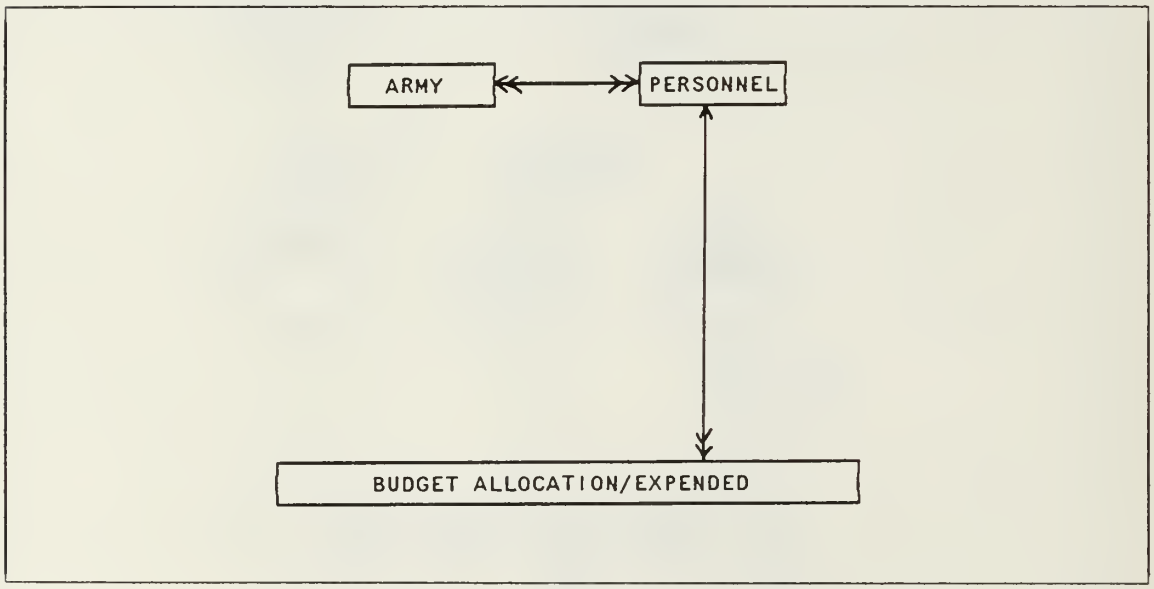

Figure 3.5 Single-Level Multi-Structure Query.

d. Multi-Level Multi-Structure Query

This is the most complicated type of query required by users. In this type of query, user view of data can go across the Budget Structure, and at one or more levels within each budget structure. The possible

combinations of levels and structures can be a very large, if not infinite in number. For example, the user might want to know the total budget for Army, Major Command X, Personnel Expense for Payroll, and for Activity type Operation. See Figure 3.6.

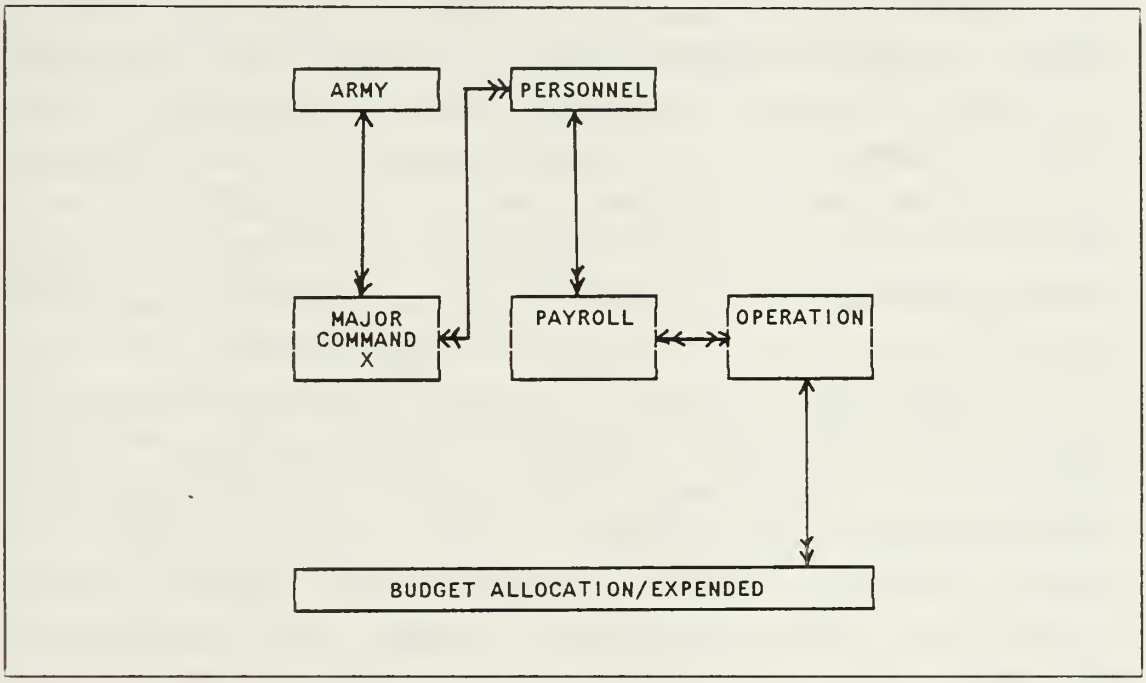

Figure 3.6 Multi Level Multi Structure Query.

The primary concern of the logical design is to provide initial knowledge to the user about the purpose for gathering their views of data. If users understand the definition of a view in order to think of data in terms of individual group or related fields, they will be able to develop their own views and develop their own queries.

2. Report Type

In principle, the report type and query type have the same views. The report types can consist of single

level single structure, multi level single structure, single level multi structure, and multi level multi structure. The major differences are that reports are usually presented in more formal format, provided in a hardcopy output, and usually contain a very large amount of information.

At the top level of organization, reports have to be produced for top management decision making, such as: a summary report of Budget Proposal arranged by Organizational Structure, Expense Structure, Program Structure, and an overall combination; a summary report for the centralized budget at the Department of Finance, centralized budget at DODS, and non-centralized budgets; a summary report budget in terms of Supervision and Control Management; and summary reports for project or non-project type budgets.

Many other kinds of reports are also produced for upper level management, middle level management, and the lowest level of management. Such reports contain information regarding budget allocation for each Branch of Service with a detailed budget for each Major Command, reports on money expended for a particular Major Expense, detailed listing of expended transactions made by Unit Commands, and etc. Names and definitions of reports produced in the current system which might have to be used to support the proposed system are listed in [Ref. 6].

3. Retrieval/Update

There are two general types of retrieval and update required by the user; these are retrieval/update during the Budget Planning Process and retrieval/update during the Budget Management Cycle. In the current system, most processes are established in batch mode. Using this mode, the processing response time is significantly slow and the users are totally isolated from the data processing

activity. Due to time constraints and tight schedules, very often users are disappointed waiting for computer outputs.

In the Budget Planning Cycle is contained the set of processes that determines which activities and projects must be done and how much money must be allocated to accomplish those activities and projects in forthcoming fiscal years. This cycle includes priority setting, determination of which units are responsible for each activity/project, and the policy of how the budget will be divided. In this cycle, the negotiation process occurs to obtain a reasonable budget for a specific organization, expense type, or activity. Data can be changed many times in the Budget Negotiation Meetings. The type of retrieval can be very similar to the four types of the Queries.

On the other hand, it is in the Budget Management Cycle where transactions including any Budget Expended by any Organization or Activity must be recorded. The updating of the Budget Record can be done by authorized personnel either at upper level management, middle management, or the lowest level management. At the same time, it must be possible for the higher level management to access or query the current status of the budget for its subordinate levels of management.

This type of retrieval involves security consideration. A resource locking facility will be provided by the proposed system. Therefore, only appropriate and authorized personnel will be able to change appropriate data. The database locking feature involves level, scope, and locking agent. Again, the retrieval/update will involve similar levels and structures as mentioned in the query type.

D. RECORD AND RELATIONSHIP DEFINITION

Two important steps occur in developing a logical design for the database system, to include the record structure definition and the logical relationship definition. Defining the record means assigning fields identified in the previous steps to a group of fields named "record". This process is an intuitive process, no precise algorithm can be used, but in most cases it is straightforward. As primary input for developing record structure, the authors make reference to the Context Diagram on Figure 3.1 which indicates the data flows in the system and the Data Dictionary listed in Appendix A that indicates the data to be kept. The authors also make reference to the Query Types, Report Types, and Retrieval/Update Requirements as consideration to determine the most efficient logical design.

Relationship is the heart of the database concept. It defines how one record type is associated with another record type. A relationship can be in the form of one-to-one, one-to-many, and many-to-many. Based on the relational model, the logical design of this prototype will only be concerned with one-to-many relationships. A one-to-one relationship can be combined into a record type, and, a many-to-many relationship can be broken into two or more one-to-many relationships. In general, one-to-many relationships convey two assertions. First, a particular record has zero, one, or more other records associated with it. On the other hand, this other record is associated with one and only one of that particular record.

In addition, the authors try to develop the record with a high degree of flexibility to anticipate any possible changes and expansion in the future. As close as possible, the design is modeled after the way in which users perceive data.

1. Normalization Forms

If we try to recognize the central thought of the Budgeting System there is an important entity named Budget which consists of Budget Allocation and Budget Expended. A Budget is a representation of amounts of money allocated and expended for a particular organization, for a particular expense type, and to support a particular activity. Therefore, we now begin our discussion of the Budget entity and go over the logical design.

a. Organization View

From the organization view a budget must contain certain attributes that indicate the properties of the Budget. These attributes include the amount of money initially allocated, the allocation that has been changed, the allocation that has already been expended, and the lowest level of the organization to which the Budget is allocated. We can simply develop the Budget Record as illustrated in Table I.

TABLE I
SIMPLEST BUDGET RECORD

```
BUDGET(Initial_Allocation, Modified_Allocation,
   Expended_1,.., Expended_n, Unit_Command)
```

In any case, the Unit_Command will not stand alone; a Unit_Command must belong to a certain Major_Command and a Certain Branch_of_Service. Therefore, we refine our

Budget record further; we will obtain the Budget record as indicated in Table II.

TABLE II

EXPANDED BUDGET RECORD

```
BUDGET(Initial_Allocation, Modified_Allocation,
   Expended_1,.., Expended_n, Unit_Command,
   Major_Command, Branch_of Service)
```

However, we know from the data dictionary that each Unit_Command, Major_Command, and Branch_of_Service consists of a code and description thus the Budget_Structure might look like Table III.

TABLE III

COMPLETED BUDGET RECORD

```
BUDGET(Initial_Allocation,  Modified_Allocation,
  Expended_1,..., Expended_n,  Unit_Command_Code,
  Unit_Command_Description,  Major_Command_Code,
  Major_Command_Description, Branch_of_Service_Code,
  Branch_of_Service_Description)
```

A record layout as shown above, in logical database design, has a consequence called modification anomalies which include insertion and deletion anomalies. These anomalies can be described as the following. In the Budget record as indicated, a fact about Unit_Command, Major_Command, and Branch_of_Service can only enter into the system if we have a fact about Budget. This restriction is

called insertion anomaly. On the other hand, we might lose some facts about Unit_Command when we delete a Budget record from the system. We will then eliminate these modification anomalies using criteria for normalization forms.

(1) First Normal Form. As a starting point, we will consider the first normal form of the Budget Record. This can be achieved by removing the repeating group and generating another record. Therefore, we remove the Budget_Expended from Budget record and generate two records Budget_Allocation and Budget_Expended as shown in Table IV.

TABLE IV

FIRST NORMALIZATION OF BUDGET RECORD
FROM THE ORGANIZATION VIEW

BUDGET_ALLOCATION (Initial_Allocation,
Modified_Allocation, Unit_Command_Code,
Unit_Command_Description, Major_Command_Code,
Major_Command_Description, Branch_of_Service_Code,
Branch_of_Service_Description)

BUDGET_EXPENDED (Transaction-identification,
Budget_Expended, Unit_Command_Code,
Unit_Command_Description, Major_Command_Code,
Major_Command_Description, Branch_of_Service_Code,
Branch_of_Service_Description)

(2) Second Normal Form. As a record identifier for both records, we can select Branch_of_Service, Major_Command, and Unit_Command as the candidate key. However, having both code and description in these records will cause a lot of redundancy. Using Branch_of_Service_Code, Major_Command_Code, and Unit_Command_Code has already uniquely identified the records. But then, this violates the second normal form. The second normal form is satisfied, if all non-key fields

are facts relating to the key fields. In this case, Branch_of_Service_Description, Major_Command_Description, and Unit_Command_Description are not facts of the record key. Therefore, we removed them from the Budget record and created other records for each of them.

(3) Third Normal Form. The last example of the record structure as illustrated in Table V has satisfied the third normal form. By definition, the record is in the third normal form if no non-key field is fact relating to other non-key field, which means no transitive dependencies exist in all of the non-key fields.

A conceptual representation of an association between records can be defined as a relationship. In the relational database model, the relationship between records is established using the field and field content within each record. In the Budget record mentioned above, we can see the relationship between Branch_of_Service and Major_Command. There is a common field belonging to both records which is the Branch_of_Service-Code. There are also common fields belong to Major_Command and Unit_Command records which are Branch_of_Service_Code and Major_Command_Code. The relationship can be one-to-one, one-to-many, or many-to-many which can be seen in the data structure diagram shown in Figure 3.7.

This inter-relationship is represented by another record called intersection record which performs as a link between two records through it common field. For example, the UNITC-MAJORC_REL is intersection record between UNIT_COMMAND record and MAJOR_COMMAND record.

(4) Fourth Normal Form. There are also in the fourth normal form since there is no multivalued dependency among the record keys. This means that no single record key contains more than one fact.

TABLE V

THIRD NORMAL FORM OF BUDGET RECORD
FROM ORGANIZATION VIEW

BRANCH_OF_SERVICE(Branch_of_Service_Code,
Branch_of_Service_Description)

MAJOR_COMMAND(Major_Command_Code, Major_Command_
Description, Location_Code)

MAJORC_BRANCH_REL(Major_Command_Code, Branch_of-Service_
Code)

UNIT_COMMAND(Unit_Command_Code, Unit_Command_Description,
Location_Code)

UNITC_MAJORC_REL(Unit_Command_Code, Major_Command_Code)

LOCATION(Location_Code, Province, City, District, Area)

BUDGET_ALLOCATION(Budget_Allocation_Code,
Initial_Budget, Modified_Budget)

BUDGET_A_UNITC_REL(Budget_Allocation_Code, Unit_Command_
Code)

BUDGET_EXPENDED(Transaction_Identification, Amount_
Expended, Date)

BUDGET_ALLOC_EXP-REL(Transaction_Identification, Budget_
Allocation_Code)

Note: ---- indicate record key

(5) Fifth Normal Form. For all records which have more than one key field, we consider the probability of violation of the fifth normal form. However,when all the records indicated above can be reconstructed from their projection, they satisfy the fifth normal form [Ref. 1].

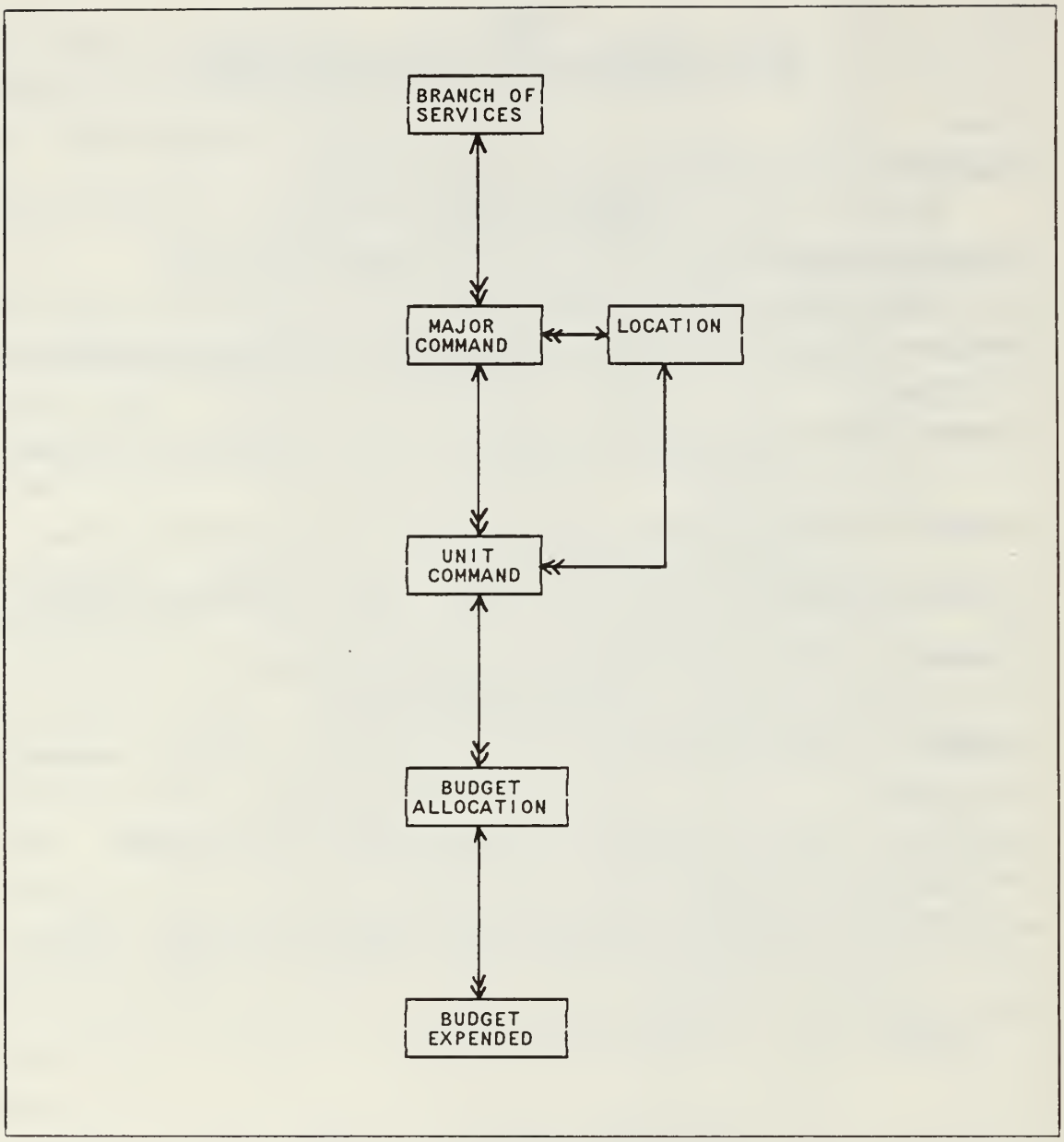

Figure 3.7 Data Structure Diagram from Organization View.

b. Internal Classification View

Similar to the Organization view, the same Budget can be viewed in terms of Internal Classification. In the Internal Classification, a Budget may be classified

TABLE VI

THE THIRD NORMAL FORM OF BUDGET RECORD
FROM INTERNAL CLASSIFICATION VIEW

BUDGET_SOURCE(Budget_Source_Code, Budget_Source_
Description)

BUDGET_TYPE(Budget_Type_Code, Budget_Type_Description)

BUDGET_TYPE_SOURCE_REL(Budget_Type_Code, Budget_Source_
Code)

FUND_TYPE(Fund_Type_Code, Fund_Type_Description)

COST_TYPE(Cost-Type_Code, Cost_Type_Description)

CONTROL_PROGRAM(Control_Program_Code,Control_Program_
Description)

SUPERVISORY_PROGRAM(Supervisory_Program_Code, Supervisory_
Program_Description)

SUPERVISORY_CONTROL_REL(Supervisory_Program_Code, Control_
Program_Code)

BUDGET_ALLOCATION(Budget_Allocation_Code,
Initial_Budget, Modified_Budget)

BUDGET_A_BUDGET_T_REL(Budget_Allocation_Code, Budget_
Type_Code)

BUDGET_A_FUND_T_REL(Budget_Allocation_Code, Fund_Type_
Code)

BUDGET_A_COST_T_REL(Budget_Allocation_Code, Cost_Type_
Code)

BUDGET_A_SUPERVR_REL(Budget_Allocation_Code, Supervisory_
Program_Code)

BUDGET_EXPENDED(Transaction_Identification, Amount_
Expended, Date)

BUDGET_ALLOC_EXP-REL(Transaction_Identification, Budget_
Allocation_Code)

Note: ---- indicate record key

into Budget Source and Budget Type, Fund Type, Cost Type, and Control and Supervisory Programs. Using the normalization form criteria we will be able to construct the logical data structure and relationship of the Budget record as shown in Table VI and the Figure 3.8.

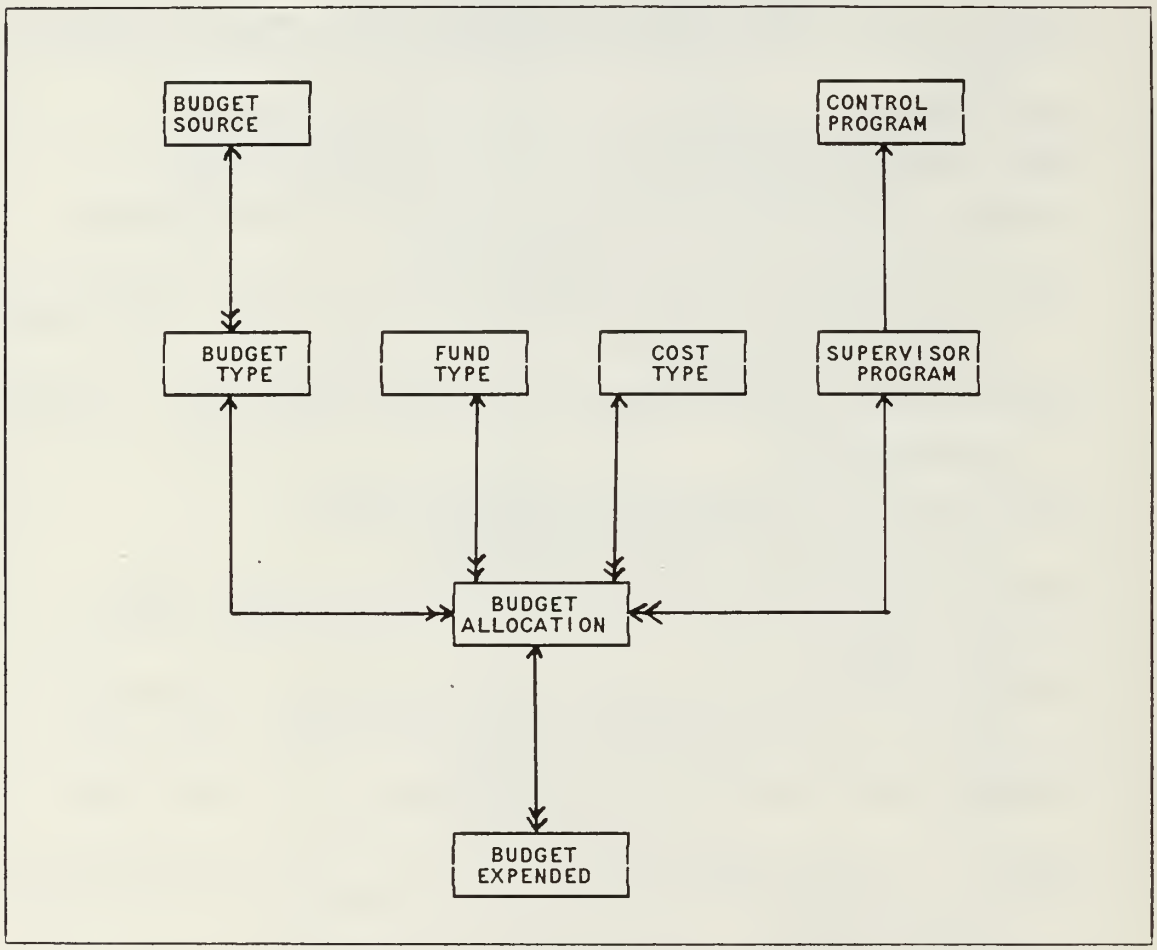

Figure 3.8 Data Structure Diagram from Internal Classification View.

c. Program Classification View

The program classification defines the Budget in terms of program and activities performed by the DODS and

TABLE VII

THE THIRD NORMAL FORM OF BUDGET RECORD
FROM PROGRAM CLASSIFICATION VIEW

MAIN_PROGRAM(Main_Program_Code, Main_Program_Description)
PROGRAM(Program_Code, Program_Description)
PROGRAM_MAIN_P_REL(Program_Code,Main_Program_Code)
ACTIVITY(Activity_Code, Activity_Description)
ACTIVITY_PROGRAM_REL(Activity_Code, Program_Code)
BUDGET_ALLOCATION(Budget_Allocation_Code,
Initial_Budget, Modified_Budget)
BUDGET_A_ACTIVITY_REL(Budget_Allocation_Code, Acitivity_
Code)
BUDGET_EXPENDED(Transaction_Identification, Amount_
Expended, Date)
BUDGET_ALLOC_EXP-REL(Transaction_Identification, Budget_
Allocation_Code)

Note: ---- indicate record key

the Armed Forces. It reflects the main functions of the DODS in establishing its mission. The program and activities of the DODS Budget are divided into three level classifications. At the first level, there is the Main Program Budget which consists of four items including Defense Forces, Security Forces, General Support, and Bhakti Abri. Every Main Program Budget consists of one or more Program Budgets. Actually, each Program Budget can consist of one or more Activities. Activity is defined as a homogeneous action performed by a collection of human resources, devices, money, and time. In the current DODS Budgeting System there are six Activities including Personnel Maintenance, Material Maintenance, Organic, Functional, Program, and Operation. If we closely examined

the Code Structure as described in [Ref. 5], we will see that there is a many-to-many relationship between the Program Budget and Activities. Therefore, we see that one activity can consist of one or more Program Budgets. Performing the Normalization of data results in obtaining the logical record structure as shown in Table VII.

d. Expense Classification View

According to Government regulations, all Official Expenses must be classified into one of the four

TABLE VIII

THE THIRD NORMAL FORM OF BUDGET RECORD FROM EXPENSE CLASSIFICATION VIEW

MAJOR_EXPENSE(Major_Expense_Code, Major_Expense_Description)
SUBMAJOR_EXPENSE(Submajor_Expense_Code,Submajor_Expense_Description)
SUBM_E_MAJOR_E_REL(Submajor_Expense_Code,Major_Expense_Code)
UNIT_EXPENSE(Unit_Expense_Code, Unit_Expense_Description)
UNIT_E_SUBM_E_REL(Unit_Expense_Code, Sub_Major_Expense_Code)
BUDGET_ALLOCATION(Budget_Allocation_Code, Initial_Budget, Modified_Budget)
BUDGET_A_UNIT_E_REL(Budget_Allocation_Code, Unit_Expense_Code)
BUDGET_EXPENDED(Transaction_Identification, Amount_Expended, Date)
BUDGET_ALLOC_EXP-REL(Transaction_Identification, Budget_Allocation_Code)

Note: ---- indicate record key

Major Expenses. These Major Expenses include Personnel, Procurement, Maintenance, and Transportation. Expense is an aggregation of an amount of money spent for material or services which are required to support an activity to reach a particular program objective.

Expense Classification consists of three levels of subdivisions which are Major Expense, Submajor Expense, and Unit Expense. Following the Normalization Criteria, we obtain the record structures as shown in Table VIII.

e. Project Identification

The final view of the Budget Structure is determining whether a Budget can be considered a Project or Non_project Budget. If a Budget is considered as a Project Budget it will be indicated by the Project Number or Project Description. A Non_project Budget has Null value in this field. The record structure then can be constructed as illustrated in Table IX.

TABLE IX

THE THIRD NORMAL FORM OF BUDGET RECORD FROM THE PROJECT VIEW

PROJECT(Project_Code, Project_Description)
BUDGET_ALLOCATION(Budget_Allocation_Code,
Initial_Budget, Modified_Budget)
BUDGET_A_PROJECT_REL(Budget_Allocation_Code, Project_Code)
BUDGET_EXPENDED(Transaction_Identification, Amount_
Expended, Date)
BUDGET_ALLOC_EXP-REL(Transaction_Identification, Budget_
Allocation_Code)

Note: ---- indicate record key

E. LOGICAL DATA STRUCTURE

From document flow and data analysis mentioned in the previous section, the authors conclude the proposed logical data structure for the DODS Budgeting System shown on Figure 3.9. The result is obtained by combining the five views of Budget Structure and refining the combined logical record structure to determine if it satisfies the normalization form.

Using the terminologies and definitions in Section D each component in the Logical Data Structure is defined as a "Relation" which can be described as a representation of a file that consists of key and non-key attributes or fields. The Logical Data Structure for the DODS Budgeting System is designed considering the elimination of the modification anomalies including the insertion and deletion anomalies. Each Relation satisfies the criteria for the normalization of data as described in the previous section. In the first normal form they are indicated by not having a repeating group. All Relations are in the second normal form because no non-key attributes defines a fact of a subset of the key-field. Every Relation is also in the third normal form. Therefore, every non-key attribute is not a fact about any other non-key attribute. They satisfy the fourth and fifth normal form due the fact that no Relation has multivalued dependencies.

The normalization form of data provides some advantages such as avoiding data integrity problems and minimizing data redundancy. In developing the structure, the authors also give consideration to obtaining the minimum performance penalty caused by normalization form.

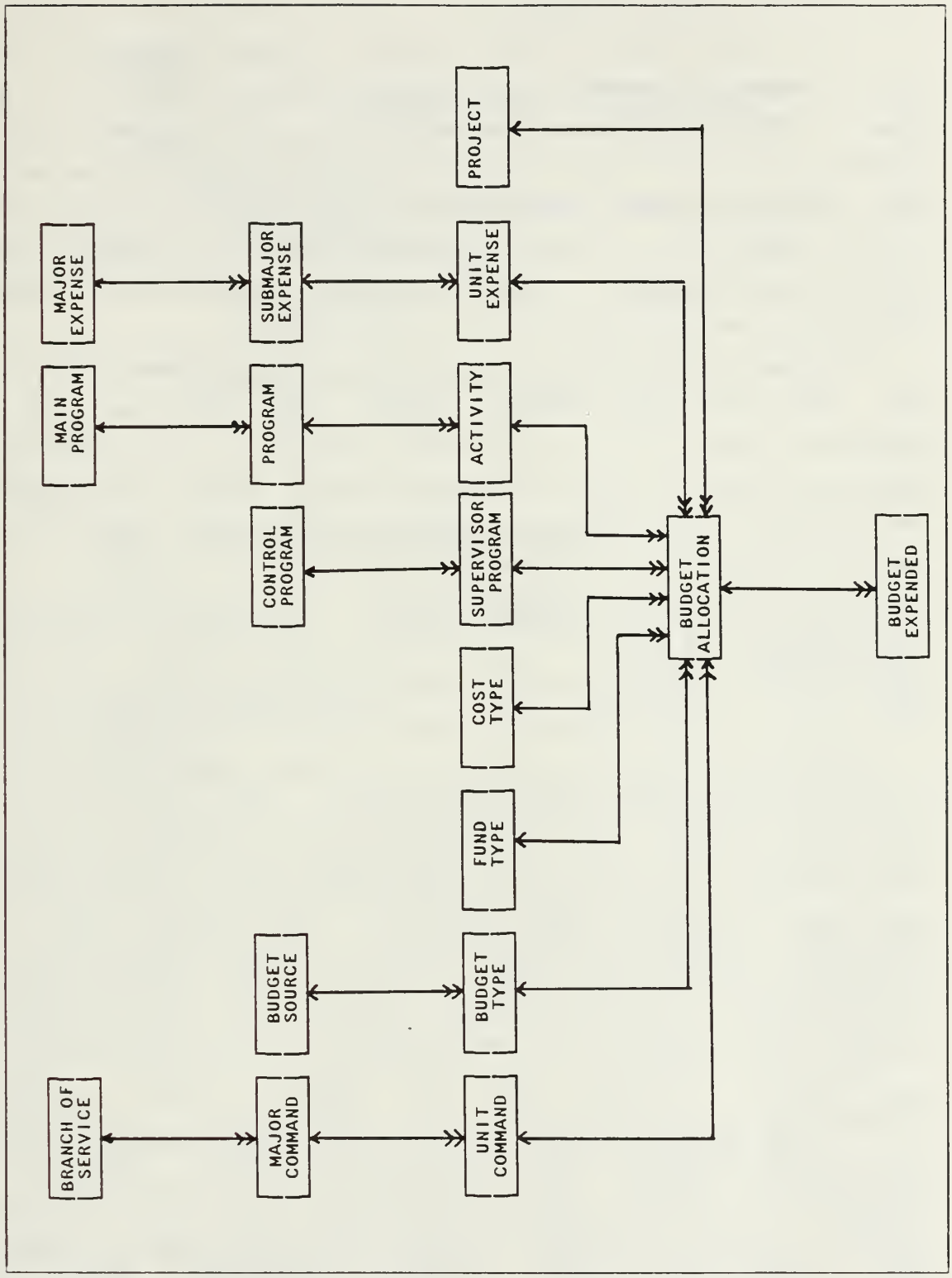

Figure 3.9 The Data Structure Diagram For the DODS Budgeting System.

F. SEMANTIC DATA MODEL

As a final result of the logical database design for the DODS Budgeting System, the authors developed a Semantic Data Model (SDM) in order to express the logical schema design. Since this is a logical thought, as far as possible we avoid the dependencies of this model on particular software such as DDL, SDM DML, and other languages. We will discuss later in the next Chapter, in the Management impact of the Logical Structure Design.

The major advantage of using the Semantic Data Model for the DODS Budgeting System Logical Data Structure is to provide a facility for expressing meaning to the Budgeting Data in order to avoid confusion among users and database developers. It provides system documentation that allows user, developer and maintenance personnel to refer to when particular problems, modifications, and enhancements of the system must be solved. The other advantage is that the model allows data to be described in context where users can see particular data from many different perspectives.

Figure 3.10 shows a sample of the SDM logical schema for a record in the DODS Budgeting System. The format is taken from [Ref. 1 p213], but some terminology has been changed by the authors in order to be consistent with the discussion set forth in the previous sections. For example, the term "record" is being used rather than the term "entity class", "field" rather than "attribute", and "key field" instead of "identifier". Actually, these words have the same meaning.

The SDM Record Name Description contains several components such as Record Name, Description, Interclass Connection, Member Fields, and Key Fields. Record Name entry is a unique name given for a particular logical record. It is an organized and identifiable aggregation of data transcribed into Database Management System.For

```
PROGRAM
   description:  The Programs under each Main
                 Program, for example: the
                 National Defense Program.
   member fields:

      PROGRAM_CODE
         value classes: CODE_OF_TASK
         mandatory
         not changeable

      PROGRAM_NAME
         value classes: NAME_OF_TASK
         mandatory

   key-field:
      MAIN_PROGRAM_CODE,PROGRAM_CODE
```

Figure 3.10 SDM Record Name Description.

example, a record name MAJOR COMMAND contains a collection of data such as MAJOR_COMMAND_CODE, MAJOR_COMMAND_DESCRIPTION, and LOCATION (of the MAJOR_COMMAND).

Description Entry contains a short narrative explanation about the record. It may provide a standard definition of a record in terms of the entity functions, characteristics, properties, and relationship with other records.

Interclass Connection Entry is given for Non-base records only and is a record constructed from subsets of other records. It names another record and specifies the characteristics of this record to be included in the new record. For example, we may have a record name SPECIAL_COMMAND which may a subset of MAJOR_COMMAND.

Member fields are fields that are logically grouped in a record. It is characterized by value class which might be defined by another record name. Value class indicates the allowable value the field has. Member fields are also

```
Record name   : SPECIAL_COMMAND
Description   : Major Command with special purpose
                and having special combat units
Interclass Connection :
                Subset of MAJOR_COMMAND which
                location code is 33000 through
                35000
Member fields : ....etc .....
```

Figure 3.11 Example of Interclass Connection Entry.

characterized by other optional field domains such as single or multivalued, value optional or mandatory, changeable or non-changeable, and overlaping or non-overlapping. An attribute is considered "single" if there is only one value and no repeating value. A member field is "optional" if null value is to be accepted and mandatory if null value is not allowed. A Non-overlapping field can specify a multivalued field and define that member of the value class that can be used once at the most.

Each record is uniquely identified by a field or concatenated fields which in our Semantic Data Model will be indicated by an entry key-field. The overall SDM Schema for the DODS Budgeting System is shown in the Appendix B.

IV. MANAGEMENT IMPACT ON THE ORGANIZATION

A. INTRODUCTION

As described in the previous chapter, the current system cannot be used any longer. The file processing method for the DODS Budgeting System is not sufficient to support the query process. Therefore, it is advisable to implement the database management system for the Budgeting System in the near future.

The database management system requires some changes in the DODS. This chapter will discuss the impact on DODS if it is decided to implement the database management system.

The major changes which will impact the organization can be described through five aspects of the database: hardware, software, data, personnel and procedures. These changes are necessary in order to ensure that the system will run properly.

B. HARDWARE

1. Computer System Hardware

No special hardware is required to run a database system. However, it does involve special programs and overhead data for the data dictionary, data pointers and data indexes [Ref. 1]. The file processing system mainly operates using the sequential access method so the data can be stored in magnetic tape or in magnetic disk in sequential mode. The database system, however, requires that all the data and programs be stored in direct access storage devices (DASD). The magnetic tape could only be used as backup storage. The current system hardware configuration in the

Data Processing Center office in DODS is comprised of the Univac 1106 main frame, with total capacity of 1 Megabyte (256 kilowords) main memory, dual central processing units, 6 tape drives, 2 printers, 1 card-reader, and 8 disk drives, each 55 megabytes (see Figure 4.1).

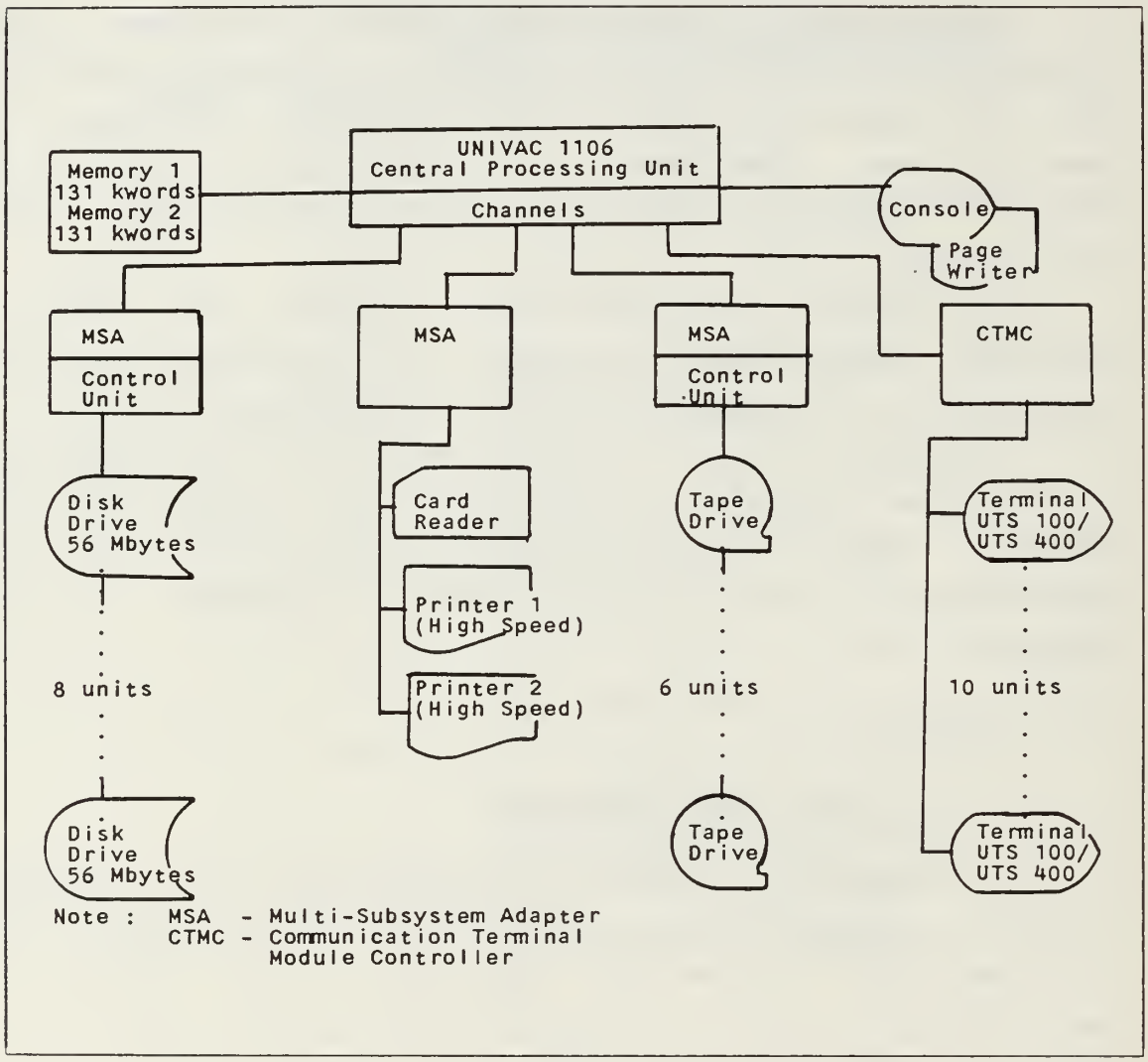

Figure 4.1 Current System Hardware Configuration.

This system configuration is sufficient to run the database management system using DMS-1100 or MAPPER-1100

(described in the next section). However, this configuration is obsolete, since this machine was installed in 1974. The maintenance cost is very high, and it is difficult to get the spare-parts. The other problem is that this system is also being utilized for general purposes. Many applications are operated by this system. The utilization of the system without any database application is close to 95%, far from the ideal utilization (about 70% to 80%). The database management system needs a lot of space in the system. Using this system, the database may occupy about 30% of the memory capacity. There will not be enough space for the other existing application system when we load the database. The current 8 disk drives are full; and, if we want to run the database, we have to exchange the disk (by using the removable disk drive). So, the current Univac 1106 system hardware configuration is not enough to run the database.

Fortunately, the DODS will install a new Univac 1100/70 system hardware configuration. This configuration will be installed at the end of 1985 and will start operations in the begining of 1986 (see Figure 4.2).

This system has more capacity than the current system; the main memory uses the cache type memory and has a maximum capacity of 16 megabytes. Two additional disk drives (250 megabytes each) will also be installed. This new system will be enough to handle the database as well as other application systems at the same time. A similar database management system for the budgeting system could also be designed to be used at other managerial levels in the DODS, such as the budgeting system for the Army, Navy, Air Force and or National Police at the branch levels. Due to the limitation of time and study this thesis will not cover this matter.

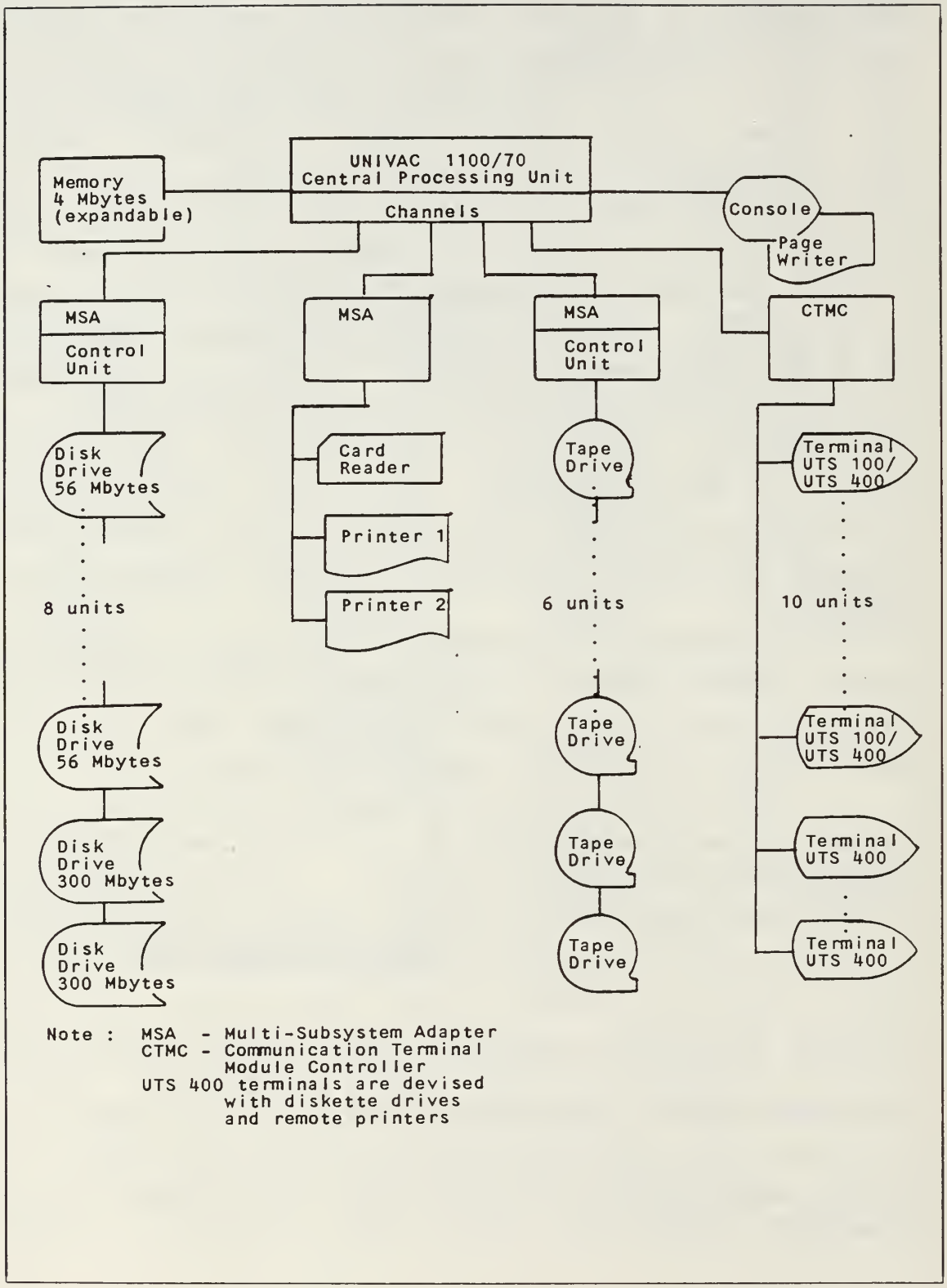

Figure 4.2 System Hardware Configuration in 1986.

2. Communication Facility and Devices

Many terminals have to be set up to make the query retrieval and query update possible in selected places. At least one portable terminal has to be set up for use during the negotiation process. Those terminals have to have the capacity to communicate on line to the main computer in the data processing center office.

It is possible to set up the communication link since the DODS already has a special communication facility throughout all the necessary offices using the DODS Satellite Communication System (KOMSAT ABRI). Data transfer from the Unit Command to Major Command, from Major Command to the Branch Office, and from Branch Office to Data Processing Center Office in DODS could also be sped up using the same communication facility. Some communication support devices such as modem (modulator demodulator) and front end processor must be purchased. To design and develop the communication system which could be used to support the budgeting system and other application systems which need the communication system, requires more detailed study. This thesis does not cover this matter.

C. SOFTWARE

1. Operating System

The Univac 1106 system used Exec-8 as the operating system. This operating system will also be used for the Univac 1100/70. The DMS-1100 and Mapper could also run under the Exec-8 operating system. So, there is no change necessary in the operating system.

2. Database Software

As mentioned earlier there are two possibilities of database software to be used.

The first is the database software DMS-1100. This database software was made to meet the standard for the Codasyl. This software is usually accompanied by other supporting database software such as DDL-1100, DML-1100, QLP-1100, and DMU-1100.

The DDL-1100 is the Data Definition Language used to define the database structure in DMS-1100 including definition of fields, records, files, record relations/sets, access method, and field and or record and or file protection. The DDL-1100 also provides features to expose the schemas and subschemas being used in the database system. The users can also use this DDL-1100 to see a particular set or subset of the database structure for their own purposes.

The DML-1100 is the Data Manipulation Language used to develop application programs. This software is imbedded into two different high level languages, ASCII-COBOL and ASCII-FORTRAN. The DML-1100 which related to the ASCII-COBOL called DML-COBOL, and to the ASCII-FORTRAN called DML-FORTRAN.

The QLP-1100 is the Query Language Processor used to help the database users who have no computer or programming background to access the database. This QLP-1100 is an unstructured language, very user friendly and can be learned by any user in minutes.

The DMU-1100 is the Data Manipulation Utility used to help the users maintain the database. This software provides some general features such as for sorting, dump, backup, recovery etc.

The second database software is the MAPPER-1100. This database software is newer than the DMS-1100. Unlike the DMS-1100, the MAPPER-1100 is the relational database software type. The relation of data is simplified by using the tabular format. This software is easier to use and more user friendly, has a high degree of flexibility for addition, elimination, and modification to any data in a record or file, and new data relation from two or more different files can easily be developed and it has windowing capability.

This software also has features to enable programmers to write applications programs through the Run Generator which can be executed either through batch mode to produce regular reports or through the demand mode to produce nonregular reports.

Backup and recovery features are provided automatically by this software and are hidden from the users to ensure the durability of the system. Any changes appear transparent to the user, and when a change has been made in a table, the system automatically records the changes in other tables, then makes the backup to the magnetic tape. The original table is located in a logical location called MAPPER0. The new table which contains the changes is recorded in the location called MAPPER1. Any changes in MAPPERn then will be recorded in MAPPERn+1. This process will continue until the user gives the UPDATE instruction. The UPDATE instruction will override the MAPPERn by the MAPPERn+1. But since the magnetic tape backup already exists, the original data can still be loaded back from the tape. The backup on the magnetic tape is also used as a backup recovery tool. If the system goes down the backup recovery tape will automatically be loaded back to the system as soon as the computer is turned on again.

The comparison between the two database softwares can be seen in Table X.

TABLE X

THE COMPARISON TABLE BETWEEN DMS-1100 AND MAPPER-1100

CRITERIA	DMS-1100	MAPPER-1100
1. Retrieval	good	good
2. Update	fair	good
3. Security	good	fair
4. Backup and recovery	fair	good
5. Ease of use	fair	good
6. Cost	--	--
7. Operating efficiency	good	fair
8. Vendor support	fair	good

Using the first criteria (Retrieval) both software have the same weight since both have features which allow the user to have direct access to retrieve the database. The MAPPER-1100 has better performance for the Updating criteria, because the updating process is transparent to the users. In the DMS-1100 updating and retrieval are separate processes. For the third criteria (Security), the DMS-1100 has better features, since the security must be done when the data structure is created. Hence, the security can be controlled and maintained centrally. In the MAPPER-1100 any users who are authorized to create data or tables could create their own data security, so the security is not centralized. In the Backup and Recovery criteria, the MAPPER-1100 has better features, because those processes are done automatically by the system, and the users will not even see those processes. In the DMS-1100 the backup and recovery processes must be done manually. From the Ease of

Use criteria, the MAPPER-1100 has better features, because the MAPPER-1100 uses the table approach. The data relation appear to the users as two dimensional tables which make it easier for users to directly see and understand the relationship among data. The Cost criteria is not evaluated because the two software are already available in the system. In the Operating Efficiency criteria, the DMS-1100 is more efficient because to operate the MAPPER-1100 at least one magnetic tape drive must be active while the MAPPER-1100 is being operated. In the Vendor Support criteria, using MAPPER-1100 is better because MAPPER-1100 is newer than the DMS-1100. From Table X then we can see that overall MAPPER-1100 performs better than DMS-1100.

3. Programs

There are two types of programs in the database system: application programs and utility programs.

The application programs are usually written by the database programmers to meet the specific requirements of the system which cannot be supported by the utility programs. The more specific the system or the more tailored the system to meet the user requirement the more application programs must be written. The application programs can be requested to be written by the database vendor, which is faster if the system specification is well described; or, they could also be written by the programmers from the Data Processing Center Office after they have had appropriate training for it. Programs written by the programmers from the Data Processing Center Office is suggested, since they will be the persons responsible for maintaining the system after the system has been developed, even though it may take longer a time to write the programs. Some of the application programs which must be written are:

a). Programs to make input of data (keyin data)

into the computer easier.

b). Programs to produce regular reports.

c). Programs to verify the input data.

d). Programs to transfer and or receive data, etc.

The utility programs are provided by either database software and or the hardware vendor. These programs provide a wide variety of services. Computer programming background is not required to use the utility programs. These programs are designed to be used by the users and or programmers to make them easier to access the database. In the DMS-1100 the utility program is supported by the QLP-1100, but in MAPPER-1100 it is already built into the database system itself.

4. Security System

A security system is necessary to ensure that only authorized users can obtain authorized data. There are at least three types of security which should exist in the database system.

The first is the security to ensure that only authorized users can get into the system. For this type of security there is usually a list of all of the users' identification number and passwords. Anyone who tries to use the system but fails to give the correct user identification and password will automatically be rejected.

The second type of security is the security to ensure that the data will only be used by the authorized user. Depending upon the function, position and classification of the person or group of persons, the users may have different levels of authorization to obtain the data. Some users may not even know the existence of other data in which they are not authorized to use.

The third is the security to ensure that only the authorized users update the data. Some users may only read the data but others may only feed the input data and some may read and update the data. So, before the database system can be used, the manager who is in charge of security for the system should determine which users are authorized to do what with which data. For budgeting system purposes we suggest that the user operators in the financial offices of those responsible for keying in the data have the authorization to feed in the data but cannot see the data which is already inside the system. The verifiers responsible for verifying data have the authorization to read and update incorrect data. The managers authorized to get the information from the system, should be divided according to managerial levels in which they are authorized only to get the information from their own data and the data of their subordinates. The DMS-1100 and MAPPER-1100 both have the system security features of those above requirements.

5. Backup and Recovery

The database software should also have features not only to control the concurrent processing but backup and failure recovery as well. The budgeting system will use the database for daily application. It is very important that the system have automatic backup as well as manual backup, both for the data and the data processing devices. In case the computer system in the data processing center malfunctions or goes off for routine maintenance, then the data processing backup will take care of that function. The backup data is important in case of catastrophic situations when data could be lost. The data backup should be carefully kept in a safe place, not too close to the original data.

If an error occurs, the system should be capable of recovering from the error automatically. The user should have no idea that an error has occured. So, to be able to do that the system should have error recovery in the data processing system and in the communication system as well. As mentioned earlier in the DMS-1100 backup and recovery are done manually, but in the MAPPER-1100 it is done automatically.

D. DATA

1. Input Data

Based on the description in Chapter 2, we know there are two types of input data in the Budget Planning Cycle. The first type is the input of data for the first initial proposal budget. This input data is from other systems such as Personnel, Accounting System, Payroll System, Logistic System etc. Those systems are already computerized. The loading process to the database can be done by using the batch mode process.

The second type of data input is the input which is given during the negotiation sessions. In the current system these transaction are processed using the batch processing mode. All transactions are collected, fed into the computer, and processed. The managers must wait until the processing is done to see the result. For the interactive system using query online this data will not be processed using the batch processing mode but must be processed using the demand processing mode. The main difference between the batch processing mode and the demand processing mode is that the batch processing mode will process the data in a file or in a group of input data transactions. If the process is interrupted then the process usually must begin from the beginning again. In the demand processing mode the data is

processed transaction by transaction. As soon as a transaction is keyed into the system the process is started and the result sent back. The user will see the result immediately after he or she has keyed in the data.

There are five different input data transactions in the Budget Management Cycle: the Authorization Letter, the Funding Note, the Bank Transfer Note, the Order of Payment Letter and the Receipt. Those five types of transactions should also be identified according to four managerial levels, the central government level, the departmental level, the branch level, the major command level and the unit command level.

In the current system those transactions are sent by mail to the data processing office and recorded into the diskette or tape. There are two recording processes for the input data. First, before the transactions are sent by the sender, and secondly, after the transactions have arrived at the destination. This activity is necessary to avoid any missing transactions during the sending process. This process may not be necessary if the transaction is transferred using the communication line. Instead of sending the hard copy transaction, the sender can send the data by using a computer via the communication line. This change is necessary to keep the database up-to-date. But this will cause another problem. Can we assume that those transactions are legal? (because there will be no legal signature or office stamp on those transactions.) If the communication is secured and the person who authorized the sending of the data is also the valid person, then we can assume that those transactions are legal.

2. Data Dictionary

As mentioned earlier the database needs a data dictionary to run the system. This data dictionary must be

maintained to keep it up-to-date. The data dictionary will also force the organization to standardize all the terminologies used in the system. This is a good benefit to the organization. But for the person who has already familiar with the old terminology for along time, it might not be easy to change. The best thing is to adopt the terminology which is already popular in the organization as the formal terminology. In case there is a deadlock as to which name should be used to describe certain data and there is no agreement between the persons who were authorized to review and decided upon a standardized terminology then the use of the alias option features in the data dictionary should be initiated.

E. PERSONNEL

1. Database Administration Personnel

Database administration personnel are a group of personnel who are assigned the task of maintaining the database responsibility [Ref. 1]. The database administration should represent the user's interest and should evolve with the database since the database is still being developed. The functions of the database personnel are:

a). Provide that the database remain standardized through the database dictionary.
b). Establish data ownership, retrieval and modification.
c). Create and disseminate recovery procedures.
d). Inform and train users how to use the database.
e). Enforce data activity policy.
f). Publish and maintain documentation.

The organization structure of the database administration personnel can be seen in Figure 4.3.

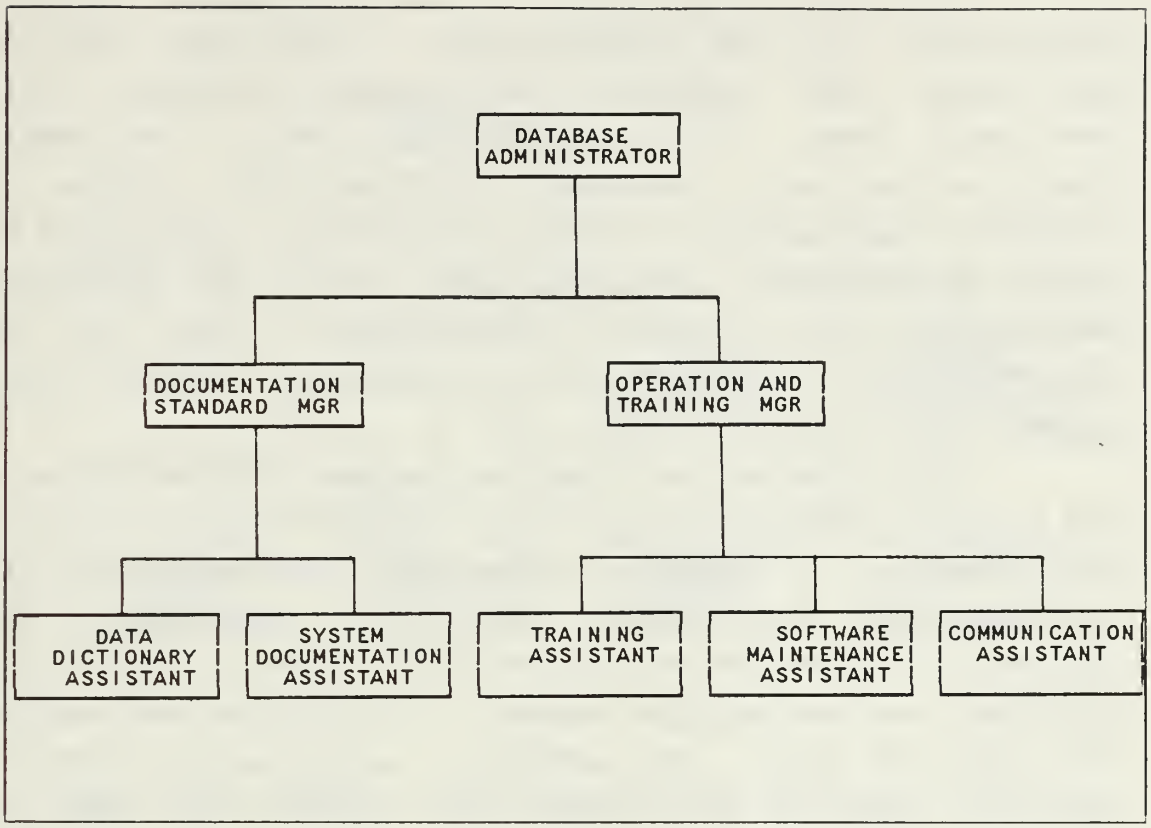

Figure 4.3 Organization Structure of the Database Administration Personnel.

The Database Administrator is a person who manages the database administration functions. This person should have considerable experience in data processing systems and a considerable experience in budgeting systems. If the manager might has difficulty in finding the right person to meet these two requirements, it would be better to train a person who has little or no knowledge of computers or about database systems but has excellent experience in budgeting, rather than train a computer specialist in the budgeting system. This statement may not be true in all cases but in most cases it is.

The Documentation and Standard Manager is a person under the Database Administrator who is responsible for

maintaining all the documentation, including the data dictionary, and publishing the necessary documents which users must know. This manager could have 2 assistants: first is the Data Dictionary Assistant who assists the manager in maintaining the data dictionary, and second the System Documentation Assistant who assists the manager in maintaining the system documentations such as the documentation of the application programs, schema and sub schema, user identification, etc.

The Operation and Training Manager is a person under the Database Administrator who is responsible for the daily performance of the database system. This manager should also be responsible for meeting the users' requirements. This manager could have three assistants: first, the Training Assistant who assists the manager in informing and training the users how to use the database, including the operators who will key in the input data transaction into the computer; and second, the Software Maintenance Assistant who assists the manager in maintaining the database software, including supervising the programmers; and the third, the Communication Assistant who assists the manager in maintaining the communication line.

The database administration function can be performed in the Data Processing Center Office as a special entity in the executive level as seen in Figure 4.4, or can be attached in the assistant level as a special assistant for the Chief of Data Processing Center as seen in Figure 4.5.

2. Programmers

The database programmers are a group of programmers who are responsible for writing and compiling the application programs for the database system. These programmers can be placed directly under the Database

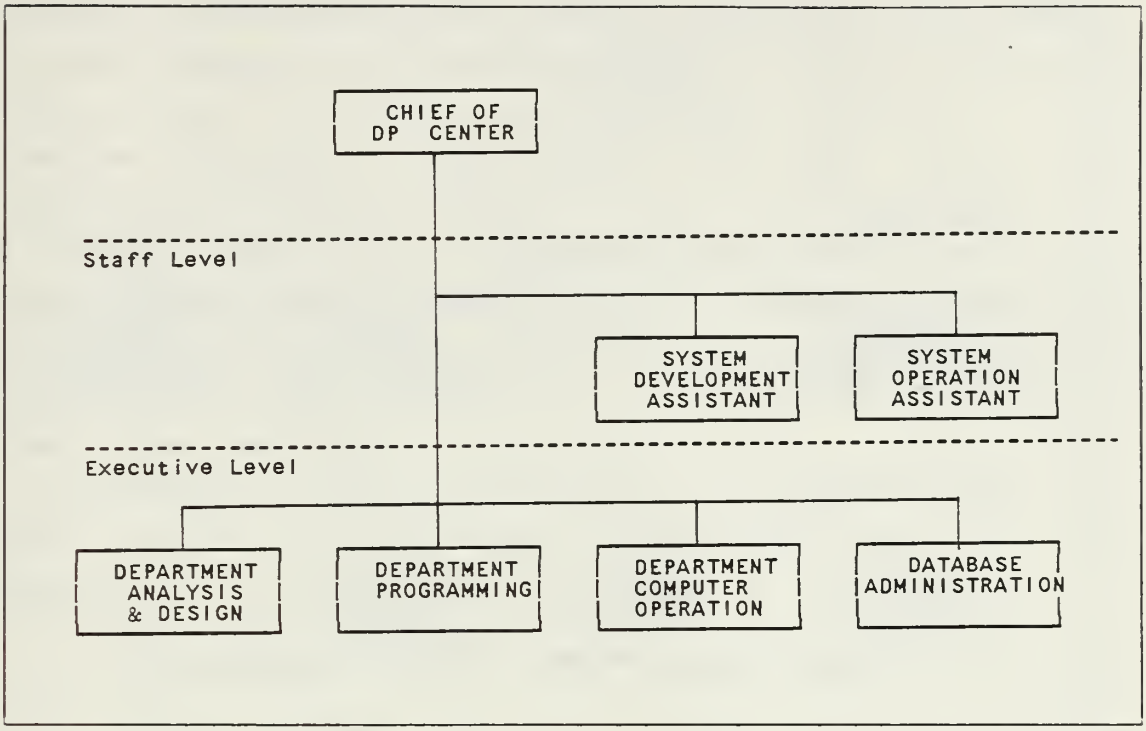

Figure 4.4 Database Administration in Executive Level.

Administrator or can be separated but supervised by the Database Administrator. Theoretically, the number of programmers required to develop the database system is fewer than the number of programmers required to develop the file processing system, because much of the processing can be done directly by the users without any help from the programmers. After the database system has been developed only two or three programmers are needed to maintain the system.

3. Users

There are two different types of users in the database system. The first are the managers who will use the database to get the information needed as a tool to help them make decisions, and the second are the operators who

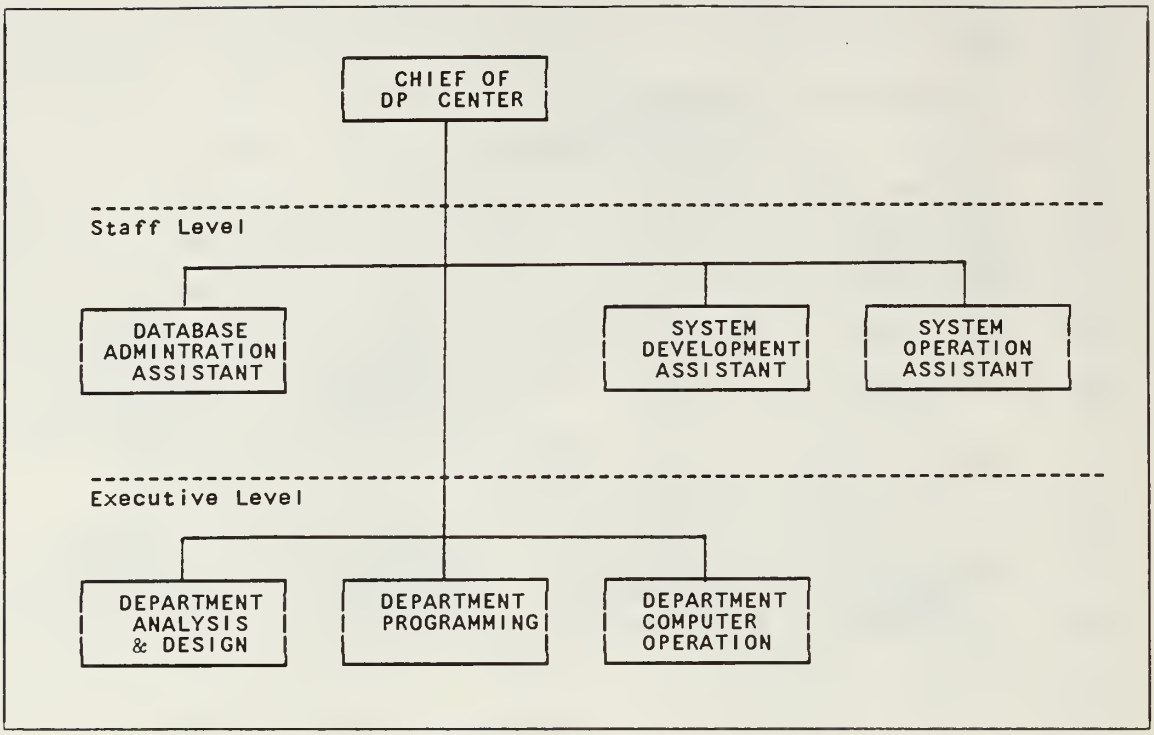

Figure 4.5 Database Administration as One of the Assistants.

are responsible for inputing data to keep the system up-to-date. Each user should have a user identification and password to ensure the security of the system. The user identification and password are given by the Database Administrator. The levels of authorization among users are not the same. Some users may be allowed only to read the database and some may be allowed to read and update the database. Some users may be authorized to access some secret data, but others may not even know that secret data exists. The data administrator must determine which user is authorized to do what to which data.

F. PROCEDURES

There are two different types of important procedures which must be developed for the Budgeting System: the

database procedure which is usually called the User Manual, and the communication procedure which is usually called the Communication Protocol. As mentioned earlier in the previous sections, this thesis will not discuss the communication matter.

The User Manual is a documented step by step instruction which gives guidance in accessing the system. Users will use the manual as a tool to help them understand and be able to access the system effectively and efficiently in terms of time, effort and money.

According to the various type of the users, the User Manual can be divided into five different types of User Manual:

1. User Manual for the Data Administrator

This manual is the most detailed system documentation which contains all the logical and physical structure of the system including the system environment, system organization, system operation, system backup and recovery, system security, system configuration, and logical, physical data structure and the data dictionary.

2. User Manual for the Managers

This manual is created to be used by the managers who will use the output of the system as a tool for a decision making. This manual contains: the description of the objectives and goals of the system, the system constraints, the system boundaries, system's input and system's output, and a general description of how the system's work.

3. User Manual for Data Entry

This manual is created to be used by the Data Entry Operator. The manual contains a step-by-step instructions of

how to key-in data into the system. This manual may also include the time scheduling for data entry and input data validation technique.

4. User Manual for On-line Data Retrieval

This manual is created to be used by any user who authorized to access the database using the on-line system facilities. The manual contains a step-by-step instruction of how to retrieve data from the database including the error messages and error handling procedures.

5. User Manual for the Batch Processing

This manual is created to be used by any user usually computer operators or maintenance programmers who will run the routine batch application programs. This manual contains the time scheduling, job control languages, and the error messages and error handling procedures.

V. COST AND BENEFIT ANALYSIS CONCEPT

A. INTRODUCTION

From the description in Chapter IV, its likely to cost more to use database processing system. Money is needed to purchase the extra hardware devices and to set up the database administration office. This chapter will discuss and guide how to make a decision on whether or not we will develop the new Budgeting System using the database.

To determine whether we should 'go' or 'not go' in order to develop a new system, we will analize two different aspects of the system. The first factor is the investment cost and maintenance cost analysis, and the second factor is the system capability analysis which are discussed in the next two sections.

B. INVESTMENT COST AND MAINTENANCE COST ANALYSIS

To develop the new Budgeting System using the database processing system mentioned in Chapter IV, we need to consider the investment costs for the additional hardware devices, communication support facilities, and for the database administration establishment.

Assume that all of this investment cost will be $(x), for the new system. The current system since it is already in operation does not need any investment cost, so the investment cost for the current system is equal to zero or $(0).

The current system uses the file processing system to operate as mentioned in Chapter I. This type of system will

make the system become data dependent and program dependent. Data dependent and program dependent means if there is any changes in data structure then the whole data must be redesigned and reloaded into the system. This will also impact the program. The program must be remodified or replaced by a new program to be able to access the data with the new data structure. The relationship between the data and the program are very tight and rigid. Every file belongs to and or is created by a certain program. Thus any changes to the data or program will effect the other.

From the experience since the Budgeting System has been computerized we know that in the beginning of every fiscal year, the data, structure of data, and the output format are changed. These changes require a lot of programming effort, and consequently require extra cost for system maintenance. In Figure 5.1 we see the changes in every fiscal year will make the maintenance cost for the current system fluctuate in the beginning of every fiscal year. Assume that the average maintenance cost of the current system is $ (y).

The new system uses database processing system. The database processing system will make the system become data independent and program independent. Any changes in the data, data structure or the report format will not impact the whole structure of data but only the part of the data structure which is changed, similar with the program, only the program which accesses that particular data will have to be modified. No data reloading as a whole. As a result the fluctuation of the maintenance cost in each fiscal year will be low for the new system as it seen in Figure 5.2.

Assumed that the average maintenance cost of the new is is $ (z), then the average cost of the investment and maintenance cost for the new system will be $ (x+z).

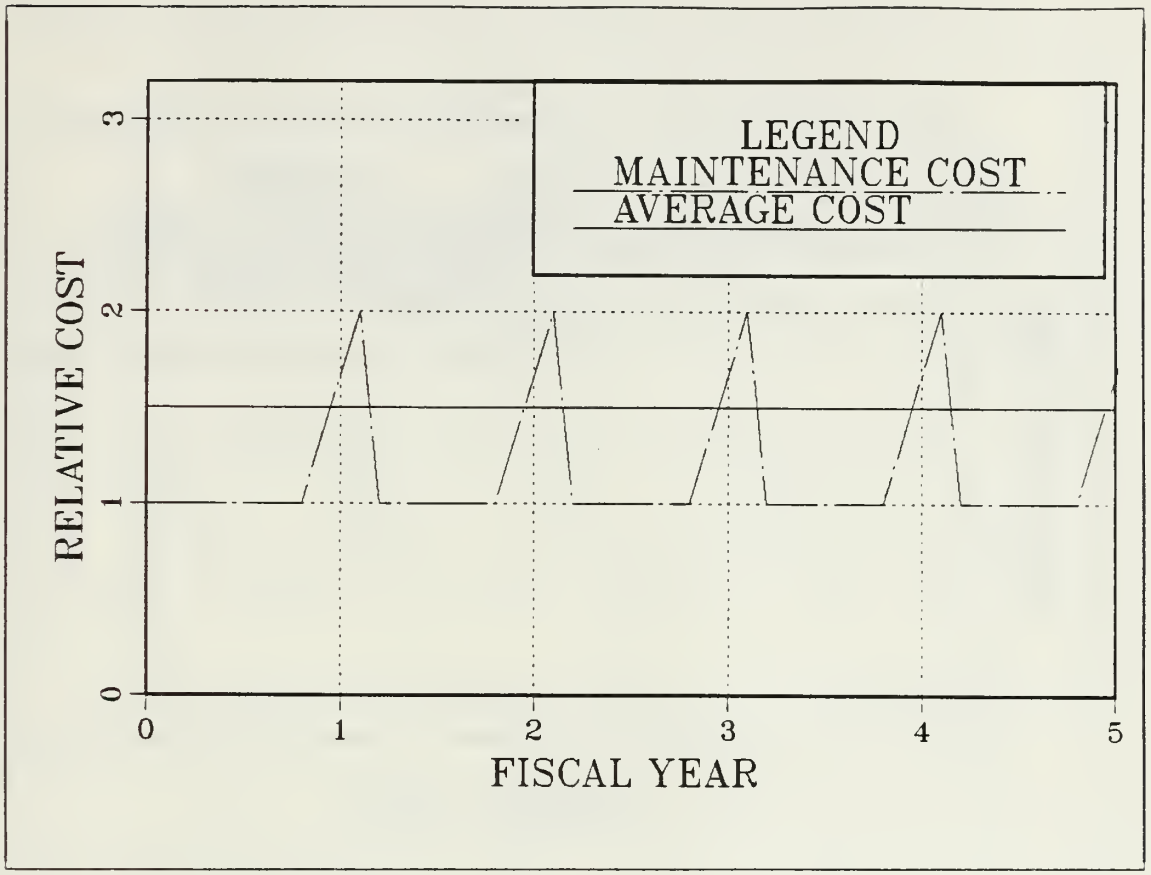

Figure 5.1 Maintenance Cost for the Current System.

If we compare the average cost of the two systems, the current system and the new system, there are two possibilities occur. See Figure 5.3. The first possibility is that the average cost of the investment cost plus the maintenance cost for the new system will be higher than the average cost of the current system or (x+z) > (y). The second possibility is that the average cost of the investment cost plus the maintenance cost of the new system will be lower than the average cost of the current system or (x+z) < (y). In the case of the second possibility the development of the new system can be started. If the first possibility occurs, then we need further analysis to

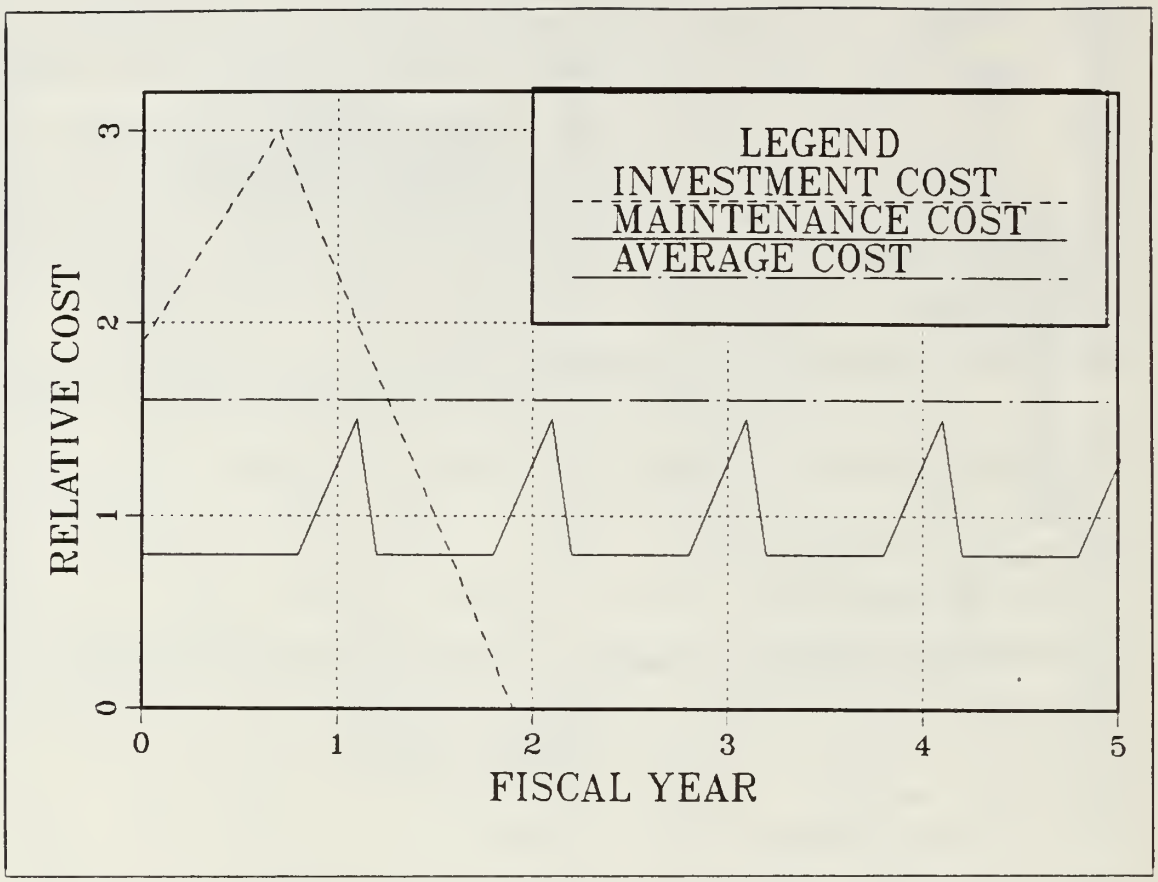

Figure 5.2 Investment and Maintenance Cost for the New System.

determine how much higher we can effort or tolerate for the cost of the new system because we also know that the new system has more capability than the current system. This analysis is discussed in the next section.

C. THE SYSTEM CAPABILITY ANALYSIS

In this section we will discuss about how much more we can afford to pay for the new system if the average cost of the new system is higher than the average cost of the current system. In doing this we must analyze the advantages and disadvantages of the two systems.

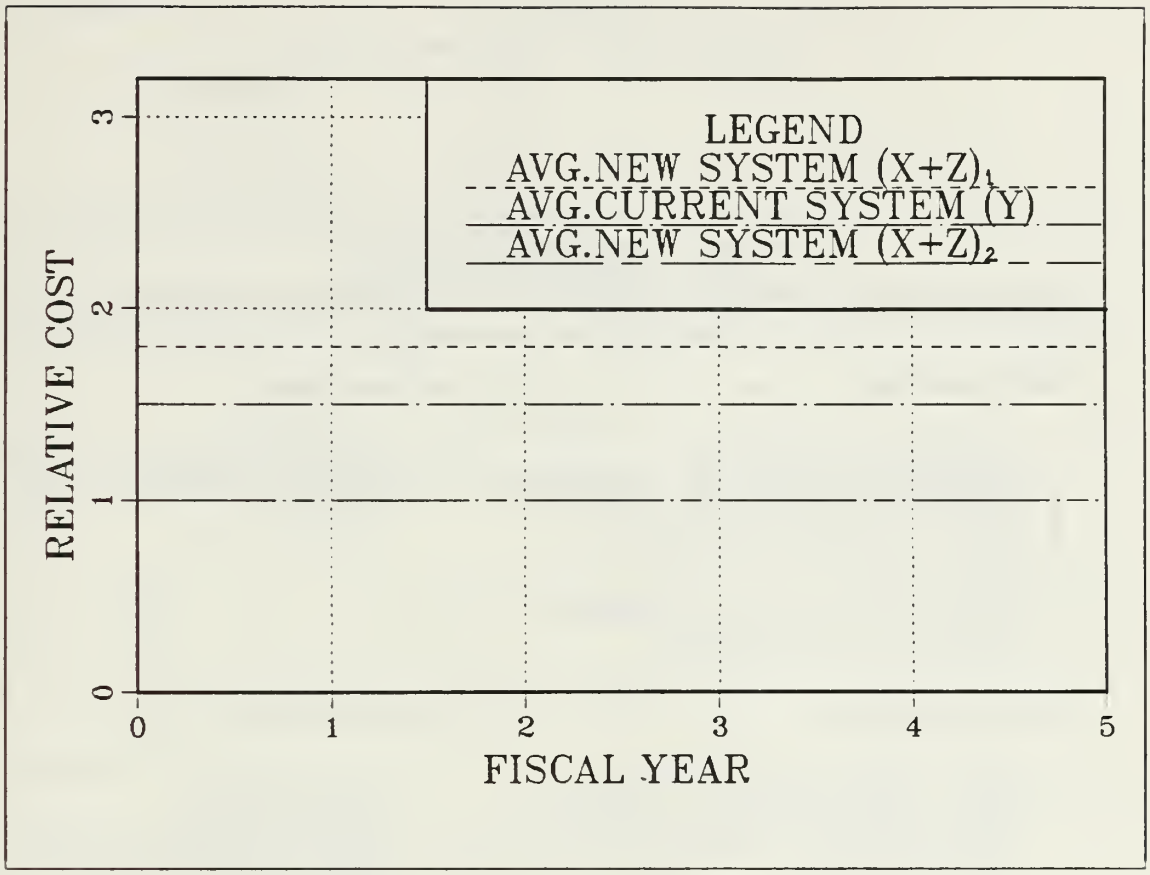

Figure 5.3 Average Cost Comparison.

Table XI shows that the new system has more advantages than the current system. If we can translate those advantages and disadvantages from each system into dollar value then we can figure out the maximum limit of the cost for the new system.

Assuming that the translation for the advantages of the new system is $ (a), the translation of the disadvantages of the new system is $ (b), the translation of the advantages of current system is $ (c), and the translation of the disadvantages of the current system is $ (d). Since we know that in the overall, the new system has more advantages than the current system, then the $ (a-b) should be greater than

TABLE XI

BENEFIT AND COST ANALYSIS

DODS BUDGETING SYSTEM	
USING DATABASE PROCESSING	USING FILE PROCESSING
Advantages	Advantages
1. Query processing 2. Easier/cheaper to maintain 3. Communication utilization 4. Data standardization 5. Less programming activity 6. Easy of use 7. More Users involvement 8. Flexible processing of order history data	1. Easier/cheaper to develop 2. Data standard is not required
Disadvantages	Disadvantages
1. Need extra hardware 2. More expensive to develop 3. Cost for database administration establishment and maintenance	1. Not ideal for query processing 2. Not flexible 3. Uncontrollable data redundancy 4. Data and program dependent 5. Require a lot of programming activity

the $ (c-d). So the maximum limit for the average cost of the new system is the average cost of the current system plus the difference of the advantages and the disadvantages of the two systems or using symbols we can formulate:

$$(X+Z)\ max = (y) + \{ (a-b) - (c-d) \}$$

D. EXAMPLE

1. Assumption

a. Maintenance cost for the current system is $ 100,000.- per year.
b. Total investment cost is $ 325,000.-
c. The maintenance cost for the new system is 40% lower than the current system.
d. The new system has a better performance by $ 50,000 per year than the current system.
e. The expected life time of the new system is five years.

2. The Cost and Benefit Analysis

a. Investment cost and maintenance cost analysis
The average cost for the new system (x+z)is:

$ (325,000/5) + $ 40/100(100,000) = $ 105,000

This figure is bigger by $ 5,000.- than the average maintenance cost for the current system, then we have to proceed to the capability analysis.

b. Capability Analysis

The new system has a better performance $ 50,000 per year, then the maximum limit of the average cost for the new system is:

$ 100,000 + $ 50,000 = $ 150,000.-

If we compare the maximum limit of the average cost for the new system and the average cost of the new system we have been calculated in the investment cost and maintenance cost analysis step, then we found that the average cost for the new system is lower than the maximum limit of the average cost by $ 45,000. So, the development of the system can be started.

VI. CONCLUSION

The implementation of the Database Management System for the DODS Budgeting System has several expectations. The DBMS provides independence between users and programs also between programs and data. It also provides an integrated system which caused significantly reduced data duplication and data redundancy, therefore, it increases the data consistency throughout the system. More information can be obtained from the systems.

The DBMS has been proven by many application systems to be considered the most cost effective application development. The cost might be high in the beginning of the implementation due to additional software and hardware configurations in order to enable the application of the database. However, in the long run, significant saving in the operational cost can be obtained.

The most important aspect obtained from the database implementation is more user involvement in the computer processing, the better access data structure, and the better control over data. Users are made responsible for the accuracy of the data processing. High level database language, such as Mapper-1100, is now available which makes database retrieval, inquiry or report generator, an easy task that can be performed by non-data-processing personnel. Users or end-users can gain access data in a simple fashion where complexity of the Mapper-1100 database itself is totally hidden from them. The English-command system of Mapper-1100 is designed for executives, middle management, and administrative personnel.

In addition, the database will be very flexible, therefore, an unanticipated request for data or output

reports can be easily handled without having to write a program that is very time consuming.

The prototype DBMS for DODS Budgeting System will provide users with the Decision Support System capability which is a very important feature used in the negotiation process of the Budget Planning Cycle. The Decision Support System is one kind of tool that a manager or user may use to create model to predict future Budget consequences in terms of predicted conditions or allowable input variables. Prototyping is a new approach to the design of some application systems that can be applied in the DODS Budgeting System. It is supported by on-line technology which make possible a direct interaction between user and system. In the continuing steps, the system allows user requirement and system design to evolve together into a destination goal of a production version of the database system. Hopefully, according to James C. Wetherbe [Ref. 3], prototyping tends to generate some benefits such as shorter development time, accurate determination of user requirements, greater user support and involvement, and a less threatening process to the user. The continual changes of the system can be continually and dynamically established when new requirements or problems arise.

As described in the previous sections, the current system should not be retained any longer. Continuing to use the file processing method for the DODS Budgeting System will cause many problems in the future. Even if this method is less expensive in the short term, in the long term there are to many disadvantages. Therefore, there are good reasons to implement the database management system for the DODS Budgeting System in the near future.

APPENDIX A
THE DATA DICTIONARY

1. Name : BRANCH_OF_SERVICE

   Description : Define branches under the DODS organization such as DODS Staff, Army, Navy, Air Force, and Police

   Format : Alphanumeric maximum 30 character length

   Coding :

Code	Name
1201	DODS Staff
1221	ABRI Headquarter
1222	Army
1223	Navy
1224	Air Force
1225	Police

   See Book III of [Ref. 5 p. 5]

   Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

   Storage : Budget Files

   Synonyms : Angkatan, Unit Organisasi

2. Name : MAJOR_COMMAND

   Description : Define Major Command under each Branch of Service.

Format : Alphanumeric maximum 30 character length

Coding : See Book III of [Ref. 5 p. 45-50]

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Komando Utama

3. Name : UNIT_COMMAND

Description : Define Unit Command under each Major Command.

Format : Alphanumeric maximum 30 character length

Coding : See Book III of [Ref. 5 Appendix 6]

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Satuan Kerja

4. Name : MAIN_PROGRAM

Description : The Primary Budget Program such as Defense Forces, Security Forces, General Supports, and Travels.

Format : Alphanumeric maximum 30 character length

Coding :

Code	Name
1	Defense Forces
2	Security Forces
3	General Supports
4	Bhakti Abri

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Program Utama

5. Name : PROGRAM

Description : The Programs under each Main Program, for example: the National Defense Program

Format : Alphanumeric maximum 30 character length

Coding : See Book III of [Ref. 5 p.51, 63-78]

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : -

6. Name : ACTIVITY

Description : The specific Activity under each Program, example:

Ship Operation

Format : Alphanumeric maximum 30 character length

Coding : See Book III of [Ref. 5 p. 52-53]

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Kegiatan

7. Name : MAJOR_EXPENSE

Description : The overall expense categories such as Personnel, Procurement, Maintenance, and Transportation

Format : Alphanumeric maximum 30 character length

Coding :

Code	Name
1	Personnel Expense
2	Procurement Expense
3	Maintenance Expense
4	Travel Expense
7	Tax / Non-tax
8	Internal DODS

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Pasal

8. Name : SUB_MAJOR_EXPENSE

   Description : The subcategories of each Major Expense, such as Payroll, Vehicle Maintenance, etc

   Format : Alphanumeric maximum 30 character length

   Coding : See Book III of [Ref. 5 p. 54-61]

   Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

   Storage : Budget Files

   Synonyms : Anak Pasal

9. Name : UNIT_EXPENSE

   Description : The Unit Expenses of each Sub Major Expense, such as Military Payroll, Armed Vehicle Maintenance

   Format : Alphanumeric maximum 30 character length

   Coding : See Book III of [Ref. 5 p.54-61]

   Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

   Storage : Budget Files

   Synonyms : Mata Anggaran

10. Name : BUDGET_SOURCE

Description : The classification of budget according to the source as described in the APBN legislation

Format : Alphanumeric maximum 30 character length

Coding :

Code	Name
10	Main Budget / Anggaran Induk (A.I.)
20	Additional Expense Budget / Anggaran Belanja Tambahan (A.B.T.)
30	Continual Program Budget / Anggaran Program Lanjutan (A.P.L.)
40	Suplission Program Budget / Anggaran Program Suplisi (A.P.S.)

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Sumber Anggaran

11. Name : BUDGET_TYPE

Description : Subclassification of the Budget Source such as Routine, Developmeny, and etc.

Format : Alphanumeric maximum 30 character length

Coding :

Code	Name
101	A.I. Routine
102	A.I. Development
201	A.B.T. Routine
202	A.B.T. Development
301	A.P.L. First Year
302	A.P.L. Second Year
303	A.P.L. Third Year
401	A.P.S. Routine

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Jenis Anggaran

12. Name : FUND_TYPE

Description : The classification of budget according to the way it is funded such as centralized or non-centralized

Format : Alphanumeric maximum 30 character length

Coding :

Code	Name
1	Centralized Fund DOF
2	Centralized Fund DODS
3	Devisa Fund

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Jenis Dana

13. Name : COST_TYPE

Description : The classification of budget according to the way it is expended such as for Research, Investment, or Maintenance

Format : Alphanumeric maximum 30 character length

Coding :

Code	Name
1	Research, Development, Measurement, and Evaluation (P3.E)
2	Investment (Ivi)
3	Motivation and Maintenance / Penggiatan dan Pemeliharaan (P & P)

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Jenis Biaya

14. Name : CONTROL_PROGRAM

Description : The classification of budget according to the Control-Program Agency

Format : Alphanumeric maximum 30 character length

Coding :

Code	Name
1	Directorate General of General Planning and Budget (RENUMGAR)
2	Assistance of General Planning (ASRENUM)

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Pengendali_Program, DALPRO

15. Name : SUPERVISOR_PROGRAM

Description : The Subclassification of the Control_Program

Format : Alphanumeric maximum 30 character length

Coding : See Book III of [Ref. 5 p.43]

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Pengawas_Program, WASPRO

16. Name : PROJECT

Description : The Classification of Budget according to type of Project or Non-Project

Format : Alphanumeric maximum 30 character length

Coding : See Book III of [Ref. 5 p.62]

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Proyek

17. Name : BUDGET_ALLOCATION

Description : The budget allocated to a particular Unit Command, a particular Unit Expense, and a particular Activities.

Format : Numeric maximum 12 digits

Coding : None

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Alokasi_Anggaran

18. Name : BUDGET_EXPENDED

Description : The amount of budget expended by a particular Unit Command, a particular Unit Expense, and a particular Activities.

Format : Numeric maximum 12 digits

Coding : None

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Anggaran_Terpakai

19. Name : LOCATION

Description : The description of Organizational location including region, county, city, and area

Format : Alphanumeric maximum 50 characters

Coding : See Book "Kode Wilayah" published Department of Communication

Where used : Force Strength, Lumpsum, Budget Proposal, Budget Allocation, and Budget Expended.

Storage : Budget Files

Synonyms : Lokasi

APPENDIX B

THE SEMANTIC DATA MODEL OF THE DODS BUDGETING SYSTEM

```
BRANCH_OF_SERVICE
   description: The five branches under DODS
                are: DODS STAFF, ARMY, NAVY,
                AIRFORCE, and POLICE.
   member fields:
      BRANCH_CODE
         value classes: CODE_OF_BRANCH
         mandatory
      BRANCH_DESCRIPTION
         value classes: DESCRIPTION_OF_ORGANIZATION
         mandatory
   key-field:
         BRANCH_CODE

MAJOR_COMMAND
   description: The Major Commands under each
                Branch of Services, such as
                KODAM I, DAERAL 2, etc.
   member fields:
      MAJOR_COMMAND_CODE
         value classes: CODE_OF_MAJOR_COMMAND
         mandatory
      MAJOR_COMMAND_DESCRIPTION
         value classes: DESCRIPTION_OF_ORGANIZATION
         mandatory
      LOCATION_CODE
         value classes: 5 digits numeric
         mandatory
   key-field:
         MAJOR_COMMAND_CODE
```

```
UNIT_COMMAND
   description: The Unit Commands come under each
                Major Command, for example:
                Batallyion 328, RI Multatuli Ship.
   member fields:
      UNIT_COMMAND_CODE
         value classes: CODE_OF_UNIT_COMMAND
         mandatory
      UNIT_COMMAND_DESCRIPTION
         value classes: DESCRIPTION_OF_ORGANIZATION
         mandatory
      LOCATION_CODE
         value classes: 5 digits numeric
         mandatory
   key-field:
         UNIT_COMMAND_CODE

MAIN_PROGRAM
   description: The Primary Budget Program such
                Defense Forces, Security Forces,
                General Supports, and Travels.
   member fields:
      MAIN_PROGRAM_CODE
         value classes: CODE_OF_MAIN_PROGRAM
         mandatory
      MAIN_PROGRAM_DESCRIPTION
         value classes: DESCRIPTION_OF_TASK
         mandatory
   key-field:
      MAIN_PROGRAM_CODE

PROGRAM
   description:  The Programs under each Main
                 Program, for example: the
                 National Defense Program.
```

```
   member fields:
      PROGRAM_CODE
         value classes: CODE_OF_PROGRAM
         mandatory
         not changeable
      PROGRAM_DESCRIPTION
         value classes: DESCRIPTION_OF_TASK
         mandatory
   key-field:
      PROGRAM_CODE

ACTIVITY
   description:  The specific Activities
                 under each program, for
                 example: Ship Operations

   member fields:
      ACTIVITY_CODE
         value classes: CODE_OF_ACTIVITY
         mandatory
      ACTIVITIES_DESCRIPTION
         value classes: DESCRIPTION_OF_TASK
         mandatory
   key-field:
      ACTIVITY_CODE

 MAJOR_EXPENSE
   description: The overall expense categories
                such as Personnel, Procurement,
                Maintenance, Transportation.
   member fields:
      MAJOR_EXPENSE_CODE
         value classes: CODE_OF_MAJOR_EXPENSE
         mandatory
      MAJOR_EXPENSE_DESCRIPTION
         value classes: DESCRIPTION_OF_EXPENSE
```

```
        mandatory
   key-field:
        MAJOR_EXPENSE_CODE

SUB_MAJOR_EXPENSE
   description: The subcategories of each
                Major Expense, such as
                Payroll, Vehicle, etc.
   member fields:
      SUB_MAJOR_EXPENSE_CODE
         value classes: CODE_OF_SUB_MAJOR_EXPENSE
         mandatory
      SUB_MAJOR_EXPENSE_DESCRIPTION
         value classes: DESCRIPTION_OF_EXPENSE
         mandatory
   key-field:
        SUB_MAJOR_EXPENSE_CODE

 UNIT_EXPENSE
   description: Unit expenses each Sub Major
                Expense, such as Military Payroll,
                Armed Vehicles, etc.
   member fields:
      UNIT_EXPENSE_CODE
         value classes : CODE_OF_UNIT_EXPENSE
         mandatory
      UNIT_EXPENSE_DESCRIPTION
         value classes : DESCRIPTION_OF_EXPENSE
         mandatory
   key-field:
      UNIT_EXPENSE_CODE

 BUDGET_SOURCE
   description:  The classification of budget
                 according to the source as
                 described in the APBN legislation
```

```
  member fields:
     BUDGET_SOURCE_CODE
        value classes : CODE_OF_BUDGET_SOURCE
        mandatory
     BUDGET_SOURCE_DESCRIPTION
        value classes : DESCRIPTION_OF_BUDGET_
                        SOURCE
        mandatory
  key-field:
     BUDGET_SOURCE_CODE

BUDGET_TYPE
  description:  Subclassification of the Budget
                Source such as Routine, Developmeny,
                and etc.

  member fields:
     BUDGET_TYPE_CODE
        value classes : CODE_OF_BUDGET_TYPE
        mandatory
     BUDGET_TYPE_DESCRIPTION
        value classes : DESCRIPTION_OF_BUDGET_
                        TYPE
        mandatory
  key-field:
     BUDGET_TYPE_CODE

FUND_TYPE
  description:  The classification of budget
                according to the way it is funded
                such as centralized or non-cent-
                ralized

  member fields:
     FUND_TYPE_CODE
        value classes : CODE_OF_FUND_TYPE
```

```
        mandatory
    FUND_TYPE_DESCRIPTION
        value classes : DESCRIPTION_OF_FUND_
                        TYPE
        mandatory
  key-field:
    FUND_TYPE_CODE

COST_TYPE
  description:  The classification of budget
                according to the way it is expended
                such as for Research, Investment, or
                Maintenance

  member fields:
    COST_TYPE_CODE
        value classes : CODE_OF_COST_TYPE
        mandatory
    COST_TYPE_DESCRIPTION
        value classes : DESCRIPTION_OF_COST_
                        TYPE
        mandatory
  key-field:
    COST_TYPE_CODE

CONTROL_PROGRAM
  description:  The classification of budget
                according to the Control-Program
                Agency

  member fields:
    CONTROL_PROGRAM_CODE
        value classes : CODE_OF_CONTROL_PROGRAM
        mandatory
    CONTROL_PROGRAM_DESCRIPTION
        value classes : DESCRIPTION_OF_CONTROL_
                        PROGRAM
```

```
        mandatory
    key-field:
       CONTROL_PROGRAM_CODE

 SUPERVISOR_PROGRAM
    description:  The Subclassification of the
                  Control_Program

    member fields:
       SUPERVISOR_PROGRAM_CODE
          value classes : CODE_OF_SUPERVISOR_PROGRAM
          mandatory
       SUPERVISOR_PROGRAM_DESCRIPTION
          value classes : DESCRIPTION_OF_SUPERVISOR_
                          PROGRAM
          mandatory
    key-field:
       SUPERVISOR_PROGRAM_CODE

BUDGET_ALLOCATION
    description: The budget allocated to
                 a particular Unit Command,
                 a particular Unit Expense,
                 and a particular Activities.
    member fields:
       BUDGET_ALLOCATION_CODE
          value classes: 6 digit numeric
          mandatory
       INITIAL_BUDGET_AMOUNT
          value classes: VALUE_OF_MONEY
          mandatory
       MODIFIED_BUDGET_AMOUNT
          value classes: VALUE_OF_MONEY
          mandatory
    key-field:
          BUDGET_ALLOCATION_CODE
```

```
BUDGET_EXPENDED
   description: The budget expended by
                a particular Unit Command,
                a particular Unit Expense,
                and a particular Activities.
   member fields:
      TRANSACTION_ID
         value classes: CODE_OF_TRANSACTION
         mandatory
      BUDGET_AMOUNT_EXPENDED
         value classes: VALUE_OF_MONEY
         mandatory
      DATE
         value classes: date format YYMMDD
         mandatory
   key-field:
         TRANSACTION_ID

MAJORC_BRANCH_REL
   description: Intersection record for
                MAJOR_COMMAND and BRANCH_OF
                SERVICE
   member fields:
      MAJOR_COMMAND_CODE
         value classes: CODE_OF_MAJOR_COMMAND
         mandatory
      BRANCH_OF_SERVICE_CODE
         value classes: CODE_OF_BRANCH
         mandatory
   key-field:
         MAJOR_COMMAND_CODE

UNITC_MAJORC_REL
   description: Intersection record for
                UNIT_COMMAND and MAJOR_COMMAND
```

```
   member fields:
      UNIT_COMMAND_CODE
         value classes: CODE_OF_UNIT_COMMAND
         mandatory
      MAJOR_COMMAND_CODE
         value classes: CODE_OF_MAJOR_COMMAND
         mandatory
   key-field:
         UNIT_COMMAND_CODE

BUDGET_TYPE_SOURCE_REL
   description: Intersection record for
                BUDGET_TYPE and BUDGET_SOURCE
   member fields:
      BUDGET_TYPE_CODE
         value classes: CODE_OF_BUDGET_TYPE
         mandatory
      BUDGET_SOURCE_CODE
         value classes: CODE_OF_BUDGET_SOURCE
         mandatory
   key-field:
         BUDGET_TYPE_CODE

SUPERVISOR_CONTROL_REL
   description: Intersection record for
                SUPERVISOR_PROGRAM and
                CONTROL_PROGRAM
   member fields:
      SUPERVISOR_PROGRAM_CODE
         value classes: CODE_OF_SUPERVISOR
         mandatory
      CONTROL_PROGRAM_CODE
         value classes: CODE_OF_CONTROL_PROGRAM
         mandatory
   key-field:
         SUPERVISOR_PROGRAM_CODE
```

```
PROGRAM_MAIN_P_REL
   description: Intersection record for
                PROGRAM and MAIN_PROGRAM
   member fields:
      PROGRAM_CODE
         value classes: CODE_OF_PROGRAM
         mandatory
      MAIN_PROGRAM_CODE
         value classes: CODE_OF_MAIN_PROGRAM
         mandatory
   key-field:
         PROGRAM_CODE

ACTIVITY_PROGRAM_REL
   description: Intersection record for
                ACTIVITY and PROGRAM
   member fields:
      ACTIVITY_CODE
         value classes: CODE_OF_ACTIVITY
         mandatory
      PROGRAM_CODE
         value classes: CODE_OF_PROGRAM
         mandatory
   key-field:
         ACTIVITY

SUBM_E_MAJOR_E_REL
   description: Intersection record for
                SUBMAJOR_EXPENSE and MAJOR_EXPENSE
   member fields:
      SUB_MAJOR_EXPENSE_CODE
         value classes: CODE_OF_SUB_MAJOR_EXPENSE
         mandatory
     MAJOR_EXPENSE_CODE
         value classes: CODE_OF_MAJOR_EXPENSE
```

```
         mandatory
   key-field:
         SUB_MAJOR_EXPENSE_CODE

UNIT_E_SUBM_E_REL
   description: Intersection record for
                UNIT_EXPENSE and SUBMAJOR_
                EXPENSE
   member fields:
      UNIT_EXPENSE_CODE
         value classes: CODE_OF_SUPERVISOR
         mandatory
      SUB_MAJOR_EXPENSE_CODE
         value classes: CODE_OF_CONTROL_PROGRAM
         mandatory
   key-field:
         UNIT_EXPENSE_CODE

BUDGET_A_UNITC_REL
   description: Intersection record for
                BUDGET_ALLOCATION and UNIT_COMMAND
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      UNIT_COMMAND_CODE
         value classes: CODE_OF_UNIT_COMMAND
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_A_BUDGET_T_REL
   description: Intersection record for
                BUDGET_ALLOCATION and BUDGET_TYPE
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
```

```
         mandatory
      BUDGET_TYPE_CODE
         value classes: CODE_OF_BUDGET_TYPE
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_A_FUND_T_REL
   description: Intersection record for
                BUDGET_ALLOCATION and FUND_TYPE
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      FUND_TYPE_CODE
         value classes: CODE_OF_FUND_TYPE
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_A_COST_T_REL
   description: Intersection record for
                BUDGET_ALLOCATION and COST_TYPE
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      COST_TYPE_CODE
         value classes: CODE_OF_COST_TYPE
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_A_SUPERV_REL
   description: Intersection record for
                BUDGET_ALLOCATION and SUPERVISOR_
                PROGRAM
   member fields:
```

```
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      SUPERVISOR_PROGRAM_CODE
         value classes: CODE_OF_SUPERVISOR
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_A_PROJECT_REL
   description: Intersection record for
                BUDGET_ALLOCATION and PROJECT
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      PROJECT_CODE
         value classes: CODE_OF_PROJECT
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_A_ACTIVITY_REL
   description: Intersection record for
                BUDGET_ALLOCATION and ACITVITY
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      ACTIVITY_CODE
         value classes: CODE_OF_ACTIVITY
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_A_UNIT_E_REL
   description: Intersection record for
                BUDGET_ALLOCATION and UNIT_EXPENSE
```

```
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      COST_TYPE_CODE
         value classes: CODE_OF_UNIT_EXPENSE
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
BUDGET_ALLOC_EXPENSE_REL
   description: Intersection record for
                BUDGET_ALLOCATION and BUDGET_EXPENDED
   member fields:
      BUDGET_ALLOCATION_CODE
         value classes: 6 digit-numeric
         mandatory
      TRANSACTION_ID
         value classes: 15 characters string
         mandatory
   key-field:
         BUDGET_ALLOCATION_CODE
```

APPENDIX C

THE ATTRIBUTES DOMAIN

```
CODE_OF_BRANCH
   interclass connection:
       subclass of STRING of 4 position
       where the value are numeric which:
       '1201', '1221', '1222', '1223', '1224', '1225'
CODE_OF_MAJOR_COMMAND
   interclass connection:
       subclass of STRING of 3 position
       where the value are numeric which:
           for MAJOR_COMMAND_CODE : '101' - '599'
CODE_OF_UNIT_COMMAND
   interclass connection:
       subclass of STRING of 5 position
       where the value are numeric which:
           for UNIT_COMMAND_CODE : '10101' - '59999'
DESCRIPTION_OF_ORGANIZATION
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
CODE_OF_MAIN_PROGRAM
   interclass connection:
       subclass of STRING of 1 position
       where the value are numeric which:
           for MAIN_PROGRAM_CODE: '1' -  '4'
CODE_OF_PROGRAM
   interclass connection:
       subclass of STRING of 2 position
       where the value are numeric which:
           for PROGRAM_CODE: '11' - '49'
CODE_OF_ACTIVITY
```

```
   interclass connection:
       subclass of STRING of 3 position
       where the value are numeric which
           for ACTIVITY_CODE: '100' - '600'
DESCRIPTION_OF_TASK
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
CODE_OF_MAJOR_EXPENSE
   interclass connection:
       subclass of STRING of 1 position
       where the value are numeric which:
           for MAJOR_EXPENSE_CODE: '1' - '4'
CODE_OF_SUB_MAJOR_EXPENSE
   interclass connection:
       subclass of STRING of 1 position
       where the value are numeric which:
           for SUB_MAJOR_EXPENSE_CODE: '11' - '49'
CODE_OF_UNIT_EXPENSE
   interclass connection:
       subclass of STRING of 4 position
       where the value are numeric which:
           for UNIT_EXPENSE_CODE: '1101' -  '4999'
CODE_OF_BUDGET_SOURCE
   interclass connection:
       subclass of STRING of 2 position
       where the value are numeric
       '10' -  '40'
CODE_OF_BUDGET_TYPE
   interclass connection:
       subclass of STRING of 3 position
       where the value are numeric
       '101' - '401'
CODE_OF_FUND_TYPE
   interclass connection:
```

```
        subclass of STRING of 1 position
        where the value are numeric
        '1' - '4'
CODE_OF_COST_TYPE
   interclass connection:
        subclass of STRING of 2 position
        where the value are numeric
        '10' -  '40'
CODE_OF_CONTROL_PROGRAM
   interclass connection:
        subclass of STRING of 1 position
        where the value are numeric '1' and '2'
CODE_OF_SUPERVISOR_PROGRAM
   interclass connection:
        subclass of STRING of 2 position
        where the value are numeric
        '11' - '49'
CODE_OF_PROJECT
   interclass connection:
        subclass of STRING of 7 position
        where the value are numeric
        '0000000' - '9999999'
TRANSACTION_ID
   interclass connection:
        subclass of STRING of 15 position
        where the value are characters
DESCRIPTION_OF_EXPENSE
   interclass connection:
        subclass of STRING of 30 position
        where the value are characters
DESCRIPTION_OF_BUDGET_SOURCE
   interclass connection:
        subclass of STRING of 30 position
        where the value are characters
DESCRIPTION_OF_BUDGET_TYPE
```

```
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
DESCRIPTION_OF_FUND_TYPE
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
DESCRIPTION_OF_COST_TYPE
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
DESCRIPTION_OF_CONTROL_PROGRAM
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
DESCRIPTION_OF_SUPERVISOR_PROGRAM
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
DESCRIPTION_OF_PROJECT
   interclass connection:
       subclass of STRING of 30 position
       where the value are characters
VALUE_OF_MONEY
   interclass connection:
       subclass of STRING of 6 position
       where the value are numeric
       (in $1000).
```

APPENDIX D

THE DATA RELATIONSHIP

BRANCH_OF_SERVICE(Branch_of_Service_Code,
Branch_of_Service_Description)

MAJOR_COMMAND(Major_Command_Code, Major_Command_
Description, Location_Code)

MAJORC_BRANCH_REL(Major_Command_Code, Branch_of-Service_
Code)

UNIT_COMMAND(Unit_Command_Code, Unit_Command_Description,
Location_Code)

UNITC_MAJORC_REL(Unit_Command_Code, Major_Command_Code)

LOCATION(Location_Code, Province, City, District, Area)

BUDGET_SOURCE(Budget_Source_Code, Budget_Source_
Description)

BUDGET_TYPE(Budget_Type_Code, Budget_Type_Description)

BUDGET_TYPE_SOURCE_REL(Budget_Type_Code, Budget_Source_
Code)

```
FUND_TYPE(Fund_Type_Code, Fund_Type_Description)
          -------------
COST_TYPE(Cost-Type_Code, Cost_Type_Description)
          -------------
CONTROL_PROGRAM(Control_Program_Code,Control_Program_
                --------------------
   Description)
SUPERVISOR_PROGRAM(Supervisory_Program_Code, Supervisory_
                    ----------------------
   Program_Description)
SUPERVISOR_CONTROL_REL(Supervisory_Program_Code, Control_
                       ----------------------
   Program_Code)
MAIN_PROGRAM(Main_Program_Code, Main_Program_Description)
             -----------------
PROGRAM(Program_Code, Program_Description)
        ------------
PROGRAM_MAIN_P_REL(Program_Code,Main_Program_Code)
                   ------------
ACTIVITY(Activity_Code, Activity_Description)
         -------------
ACTIVITY_PROGRAM_REL(Activity_Code, Program_Code)
                     -------------
MAJOR_EXPENSE(Major_Expense_Code, Major_Expense_
              ------------------
   Description)
SUBMAJOR_EXPENSE(Submajor_Expense_Code,Submajor_Expense_
                 --------------------
   Description)
SUBM_E_MAJOR_E_REL(Submajor_Expense_Code,Major_Expense_
                   --------------------
   Code)
```

UNIT_EXPENSE(Unit_Expense_Code, Unit_Expense_Description)

UNIT_E_SUBM_E_REL(Unit_Expense_Code, Sub_Major_Expense_
Code)

PROJECT(Project_Code, Project_Description)

BUDGET_ALLOCATION(Budget_Allocation_Code,
Initial_Budget, Modified_Budget)

BUDGET_A_UNITC_REL(Budget_Allocation_Code, Unit_Command_
Code)

BUDGET_A_BUDGET_T_REL(Budget_Allocation_Code, Budget_
Type_Code)

BUDGET_A_FUND_T_REL(Budget_Allocation_Code, Fund_Type_
Code)

BUDGET_A_COST_T_REL(Budget_Allocation_Code, Cost_Type_
Code)

BUDGET_A_SUPERVR_REL(Budget_Allocation_Code, Supervisory_
Program_Code)

BUDGET_A_PROJECT_REL(Budget_Allocation_Code, Project_Code)

BUDGET_A_ACTIVITY_REL(Budget_Allocation_Code, Acitivity_
Code)

BUDGET_A_UNIT_E_REL(Budget_Allocation_Code, Unit_Expense_

Code)

BUDGET_EXPENDED(Transaction_Identification, Amount_

Expended, Date)

BUDGET_ALLOC_EXP-REL(Transaction_Identification, Budget_

Allocation_Code)

Note: ---- indicate record key

LIST OF REFERENCES

1. Kroenke, David, Database Processing, Sciences Research Associates, Inc. 1983, 1977.

2. Wetherbe, James C., Systems Analysis and Design, Traditional, Structured, and Advanced Concepts and Techniques, West Publishing Company, 1984, 1979.

3. Dolk, Daniel, Database Management System (IS 4183) Course, Naval Postgraduate School, 1985

4. Husein, A., Drs, Pokok-pokok Anggaran Negara (Extracts of Government Budgeting), CV Eko Jaya, Jakarta, Indonesia - 1984

5. The DODS of the Republic of Indonesia, Amanat Anggaran Dephankam 1984/1985, (DODS Budgeting Statement 1984/1985) Jakarta, Indonesia, May 1984.

6. The DODS of the Republic of Indonesia, Pedoman Pelaksanaan Sistim Spesifikasi Komputerisasi Data Duk/Dik Dephankam/ABRI, (Department of Defense and Security Computerize System Specification Execution Guidelines), Jakarta, Indonesia, Aug 1984

BIBLIOGRAPHY

Alan Walter Steiss, Public Budgeting and Management, DC Heath and Company, Lexington, 1972

Astrahan,M. M., Relational Approach to Database Management, ACM Trans Database Syst,1,2(1976),97-137

Baker, F.T., Chief Programmer Team Management of Production Programming, IBM Syst. J, 11,I(1972), 56-73

Benjamin, R.I., Control of the Information System Life Cycle, Wiley, New York, 1971

Bernstein P. Rothnie, Introduction to a System for Distributed Databases, ACM Trans Database Syst,5,1(1981),1-17

Boland, R.J., Jr., The Process and Product of System Design, Manage Sci. 24,9(May 1979), 887-895

Bostrom, R.P. and Heinen, J.S., MIS Problem and Failures: A socio-technical perspective (2 parts), Manage. Inf. Syst.24,9,(May 1979), 887-895

Brooks, F.T., The Mythical Man-Month: Essays on S/W Engineering, Addison-Wesley, Reading, Mass. 1979

Bubenko, J., Yr., Computer-Aide, Information Analysis and Design, Student-literature, Lund, Sweden. A report of research in Scandinavia on computer-aided analysis and design

Chamberlin,D.D., A History and Evaluation of System Requirement, ACM 24,10(1981),632-646

Chen,P. P., The Entity-Relationship Model. Toward a Unified View of Data, ACM Trans. Database Syst,1,1(1976),9-36

Chu,W., Performance of the File Directory Systems for Databases in Start and Distributed Network, NCC AFIPS Press, Arlington, Va, 1976, pp.577-587

Cleland, D.I. and King, W.R., Systems Analysis and Project Management (2nd ed.), McGraw-Hill, New York, 1975

Cooper, R.B., Management Information Requirement Assessment: The State of Art, Database (Fall 1979), 5-15

Couger, J.D., Evolution of Business Systems Analysis Techniques, Wiley, New York, 1982

Date, C., An Introduction to Database Systems, Addison Wesley, Reading, Mass., 1975

Davis, G.B., Strategies for Information Requirement Determination, IBM Syst. J.21,I(1982), 4-30

DDSWP, The British Computer Society Data Dictionary Systems Working Party report, Database (ACM SIGBDP), 9,2(Fall 1977)

DeMarco, T., Structured Analysis and System Specification, Prentice-Hall, Englewood Cliffs. N.J.,1979

Dickson, G,W., Senn,J.A. and Chervanay, N.L., Research in Management Information Systems: The Minnesota Experiments, Manage. Sci. 23,9(May 1977), 913-923

Everest, G, Managing Corporate Data Resources Objectives and a Conceptual Model of DBMS, Ph.D. Dissertation, Univ. Pennsylvania, Philadelphia, University Microfilms # 74-22,836

Fry, J and Leorey, R, The logical Record Access Approach to Database Design, ACM Comput. Surv.12, 2(1980), 179-212

Gane, C. and Sarson, T., Structured Systems Analysis: Tools and Techniques, Prentice-Hall, Inc, Englewood Cliffs, N.J.1979

Gibb, T. and Weinberg, G.M., Humanized Input, Winthrop, Cambridge, Mass., 1977

Hartman W., Matthes, H. and Proeme, A., Management Information System Handbook, McGraw-Hill, New York, 1968

IBM Corp, Business Systems Planning: Information Systems Planning Guide, Doc.GE20-8075-0, White Plans, N.Y. 1978

IBM Corp, Study Organization Plan Documentation Techniques, Tech. Rep. C20-8075-0, 1963, The system analysis and design approach advocated by IBM

Kent, W, Data and Reality, North Holland, Amsterdam, 1978

Kim, W, Relational Database Systems, ACM Comput. Surv.11, 3((1979), 185-212

King, W.R. and Cleland, D.I., The Design of Management Information Systems: An information analysis approach, Manage. Sci. 22, 3(1975)

King, W.R. and Cleland, D.I., Manager-analyst teamwork in MIS, Bus. Horizons 14,2(1971), 59-68

Kling, R., The Organizational Context of User-oriented Software Design, Manage. Inf. Syst. Q. (june 1977), 55-67

Lefkovits, H, Data Dictionary Systems, QED Information Sciences, Wellesley, Mass, 1977

Lomax, J., Data Dictionaries/Directories, IBM Syst. J.12,4(1973), 332-350

Lundeberg, M., Goldkule, G., and Nilsson, A.A., A systematic approach to information systems developments (2 parts), Inf. Syst. (1979)

Martin, J., Computer Database Organization, Prentice-Hall, Englewood Cliffs, N.J., 1977

Martin, J., Managing the Data-Base Environment, Prentice Hall, 1983

McLean, E., End users as Application Developers, Manage. Inf. Syst. Q. 3,4(1979), 37-46

Mumford, E., Computer Systems in Work Design: The Method, Wiley, New York, 1979

Munro, M.C. and Davis, G.B., Determining Management Information Needs: A Comparison of Methods, Manage. Inf. Syst. (June 1977), 55-67

Powers, Adam, and Mills, Computer Inf. Systems Development Analysis and Design, South-Western Publishing Co, 1984

Rubin, M., Introduction to the System Life Cycle, Auerbach, Pennsauken. N.J. 1970

Shaw, J.C. and Atkins, W., Managing Computer Systems Projects, McGraw-Hill, New York, 1970

Smith, J.M. and Smith, D.C.P., Database Abstractions: Aggregation and generalization, ACM Trans. Database Syst. 2,2(1977), 105-133

Sockut, G.H. and Goldberg, R.P., Database Reorganization: Principles and Practices, ACM Comp. Syst, 11,4(1979), 371-396

Stonebraker, M., Operating System Support for DBMS, Comm. ACM 24,7(1981), 412-418

Stonebraker, M., The Design and Implementation of Ingres, ACM Trans. Database Syst.1,3(1976), 189-222

Taggart, W.M., Jr. and Tharp, M.O., Dimensions of Information Requirements Analysis, Data Base 7,I (Summer 1975), 5-13

Taggart, W.M., Jr. and Tharp, M.O., A Survey of Information Requirements Analysis Techniques, ACM Comput. Surv. 9, 4(Dec 1977), 273-290

Teichroew, D. and Hershey, E., PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis, IEEE Trans. Softw. Eng. SE-3, 1(Jan 1977)

Tsichritzis, D. and Lochovsky, F., Database Management Systems, Academic Press, New York, 1977

Ullman, J., Principles of Database Systems, Computer Sciences Press, Woodland Hills, Calif 1980

Van Court, H., Jr., Systems Analysis: A Diagnostic Approach, 1967

Verhovstad, J.S.M., Recovery Techniques for Database Systems, ACM Comput. Surv, 10,2(1978), 167-196

Weinberg, G.M., The Psychology of Computer Programming, Van Nostrand Reinhold, Florence, Ky 1971

Wiederhold, Gio, Data Base Design, McGraw-Hill Co, 1983

Yourdon, E., How to Manage Structured Programming, Yourdon, New York, 1976

Zelkowitz, M.V., Perspectives in Software Engineering, Comput. Surv. 10, 2(June 1978), 197-216

Zolliker, M.L., Proceedings of a conference on application development systems, Data Base 11,3(Winter Spring 1980)

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
2.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
3.	Department Chairman, Code 54 Department of Administrative Science Naval Postgraduate School Monterey, California 93943-5100	1
4.	Computer Technology Department, Code 37 Department of Administrative Science Naval Postgraduate School Monterey, California 93943-5100	1
5.	Prof. Michael P. Spencer, Code 54Xq Department of Administrative Science Naval Postgraduate School Monterey, California 93943-5100	2
6.	Prof. Richard A. McGonigal, Code 54Mb Department of Administrative Science Naval Postgraduate School Monterey, California 93943-5100	1
7.	Kepala BPPIT Hankam Jl. RS Fatmawati - Pondok Labu, Jakarta (12450), Indonesia	1
8.	Kapus Pemasaran BPPIT Hankam Jl. RS Fatmawati - Pondok Labu, Jakarta (12450), Indonesia	1
9.	Karo Pullahta Setjen Dephankam Jl. RS Fatmawati - Pondok Labu, Jakarta (12450), Indonesia	1
10.	MayGen TNI Abdul Kadir Prawiraatmadja Jl. Lamandau III No.1 - Kebayoran Baru Jakarta Selatan, Indonesia	1
11.	Cpt Mohammad Subekti Kompl. Pullahta G7 - Pondok Labu, Jakarta (12450), Indonesia	1
12.	Cpt Widhya B. Prawiraatmadja Kompl. Pullahta G14 - Pondok Labu, Jakarta (12450), Indonesia	1
13.	Universitas Pembangunan Nasional Jl. RS Fatmawati - Pondok Labu, Jakarta (12450), Indonesia	1
14.	Kadis Pullahta TNI-AD Jl. Veteran No 5, Jakarta Pusat, Indonesia	1

15. Kadis Pullahta TNI-AL
Wisma Lumba-Lumba
Jl. Gatot Subroto 12 - Senayan,
Jakarta Pusat, Indonesia — 1

16. Dirjen Renumgar Hankam
Jl. Medan Merdeka Selatan No 7
Jakarta Pusat, Indonesia — 1

17. Dir Anggaran SKU Hankam
Jl. Medan Merdeka Selatan No 7
Jakarta Pusat, Indonesia — 1

18. Danpus Infantri TNI-AD
Jl. Wage Rudolf Supratman
Bandung, Indonesia — 1

19. Kadis Diklat TNI-AL
Jl. Gunung Sahari No 65
Jakarta Pusat, Indonesia — 1

20. Ltcol Syafei Djamil
1121 Seventh Street
Monterey, California 93940 — 1

21. Cpt Phutut H. Subroto
1121 Seventh Street
Monterey, California 93940 — 1

22. Lt Diane Gifford
340 Regina Drive North
Largo, Florida 33540 — 1

23. Cpt Park In Seop
98-313 Shinlim-2 Dong,
Kwan-Ak Gu, Seoul, Korea — 1

Thesis
S85817
c.1

Subekti
A prototype database management system for the Bugeting System of the Department of Defense and security of the Republic of Indonesia.